

PROPUESTA DE INCLUSIÓN DE LA METODOLOGÍA SCRUM AL PENSUM
DE INGENIERÍA DE SISTEMAS Y COMPUTACIÓN EN LA UNIVERSIDAD
TECNOLÓGICA DE PEREIRA

GUSTAVO ALBERTO SALAZAR JARAMILLO

CHRISTIAN CAMILO ARBELAEZ PINO

Universidad Tecnológica de Pereira – UTP
Facultad de Ingenierías
Ingeniería de Sistemas y Computación
Pereira, Junio de 2015

PROPUESTA DE INCLUSIÓN DE LA METODOLOGÍA SCRUM AL PENSUM
DE INGENIERÍA DE SISTEMAS Y COMPUTACIÓN EN LA UNIVERSIDAD
TECNOLÓGICA DE PEREIRA

GUSTAVO ALBERTO SALAZAR JARAMILLO

CHRISTIAN CAMILO ARBELAEZ PINO

Proyecto para optar al título de Ingeniero de Sistemas y Computación

Director de Proyecto

ÁNGEL AUGUSTO AGUDELO ZAPATA

Universidad Tecnológica de Pereira – UTP

Facultad de Ingenierías

Ingeniería de Sistemas y Computación

Pereira, Junio de 2015

AGRADECIMIENTOS

Agradecimiento muy especial a nuestro director de proyecto al Ingeniero Ángel Augusto Agudelo Zapata, al evaluador el ingeniero Carlos Alberto Ocampo Sepúlveda, por todos los aportes hechos a este proyecto de grado, a los docentes de Ingeniería de Sistemas y Computación de la Universidad Tecnológica de Pereira por sus contribuciones en las entrevistas y su gran acogida a la idea de este proyecto con lo que se busca aportar a la enseñanza de todos los estudiantes del programa, con lo que se espera ayudar a mejorar el trabajo en equipo en desarrollo de software.

CONTENIDO

pág.

INTRODUCCIÓN.....	12
1. DEFINICIÓN DEL PROBLEMA.....	13
1.1 ANTECEDENTES DEL PROBLEMA.....	13
1.2 FORMULACIÓN DEL PROBLEMA.....	14
2. JUSTIFICACIÓN.....	16
3. OBJETIVOS.....	19
3.1 OBJETIVO GENERAL.....	19
3.2 OBJETIVOS ESPECÍFICOS.....	19
4. MARCO REFERENCIAL.....	21
4.1 MARCO HISTÓRICO.....	21
4.2 MARCO CONCEPTUAL.....	26
4.2.1 Metodología Ágil.....	26
4.2.2 Metodología Ágil Scrum.....	27
4.3 MARCO TEÓRICO.....	28
4.4 MARCO GEOGRÁFICO.....	29

5. DISEÑO METODOLÓGICO.....	30
5.1 UNIVERSO O POBLACIÓN.....	30
5.2 MUESTRA.....	30
5.3 DEFINICIÓN DE HIPÓTESIS.....	33
5.4 VARIABLES E INDICADORES.....	33
5.4.1 Definir líneas de profundización en los estudiantes.....	33
5.4.2 Eficiencia en desarrollo de software.....	33
5.4.3 Desempeño en desarrollo de software.....	33
5.4.4 Control en desarrollo de software.....	33
5.5 INSTRUMENTOS DE RECOLECCIÓN DE INFORMACIÓN.....	33
5.6 PLAN DE ANÁLISIS DEL ESTUDIO PILOTO.....	34
5.6.1 Encuesta que se realizó a los estudiantes activos de sexto semestre a último semestre de la Universidad Tecnológica de Pereira de Ingeniería de Sistemas y Computación.....	34
5.6.2 Encuesta que se realizó a los estudiantes egresados de ISC-UTP.....	36
6. ANÁLISIS DE LAS CARACTERÍSTICAS PRINCIPALES DE LA METODOLOGÍA SCRUM PARA CONOCER SU FORMA DE TRABAJO, SUS CICLOS DE DESARROLLO Y LOS VALORES QUE OFRECE AL HACER DESARROLLO DE PROYECTOS DE SOFTWARE EN EQUIPO A TRAVÉS DE DIFERENTES ROLES.....	38
6.1 POR QUÉ UTILIZAR METODOLOGÍAS DE GESTIÓN DE PROYECTO...	40
6.2 PRESENTACIÓN DE DOS PRIMOS: CASCADA Y ÁGIL.....	40
6.2.1 Cascada.....	41
6.2.2 Ágil.....	42

6.3 SIMILARES PERO CON DIFERENCIAS.....	43
6.4 SCRUM.....	48
6.4.1 La estructura de trabajo Ágil.....	49
6.4.2 ¿Qué es Scrum?.....	50
6.4.3 La estructura Scrum.....	52
6.4.3.1 El equipo Scrum.....	53
6.4.3.2 Roles de Scrum.....	54
6.4.3.3 Eventos Scrum.....	64
6.4.3.4 Artefactos de Scrum.....	72
6.4.3.5 Reglas de Scrum.....	79
6.5 VIVIR CON SCRUM, LOS CASOS DE USO.....	81
6.5.1 ¿Que es una historia de usuario?.....	82
6.6 ¿PORQUE SCRUM?.....	83
6.6.1 Ajuste de expectativas.....	84
6.7 SCRUM EN ACCIÓN.....	85
6.7.1 La construcción del equipo.....	85
6.7.1.1 Mentalidad individual.....	86
6.7.1.2 Creación y gestión del equipo.....	88
6.7.2 Consejos Scrum exitosos.....	90
6.7.2.1 Afinando los consejos.....	91
6.8 SCRUM PROFESSIONAL.....	92
6.8.1 Certificado Scrum master (CSM).....	93
6.8.2 Certificado Scrum Product owner (CSPO).....	93
6.8.3 Certificado Scrum Developer (CSD).....	93

6.8.4 Certificado Scrum Professional (CSP).....	94
6.8.5 Certificado Scrum trainer (CST).....	94
6.8.6 Certificado Scrum Coach (CSC).....	94
 7. CONSULTA A TRAVÉS DE ENCUESTA A ESTUDIANTES DE SEXTO SEMESTRE EN ADELANTE DEL PROGRAMA DE INGENIERÍA DE SISTEMAS Y COMPUTACIÓN DE LA UNIVERSIDAD TECNOLÓGICA DE PEREIRA ALGUNAS PREGUNTAS QUE PERMITEN CONCLUIR LA NECESIDAD DE INCLUIR AL PENSUM LA METODOLOGÍA ÁGIL SCRUM PARA MEJORAR EL TRABAJO EN EQUIPO.....	96
7.1 RESUMEN ENCUESTA DE ESTUDIANTES ACTIVOS.....	96
7.1.1 Análisis de la presente encuesta.....	97
7.2 RESUMEN ENCUESTA DE EGRESADOS.....	100
7.2.1 Análisis de la presente encuesta.....	101
7.3 ANÁLISIS DE LA COMPARACIÓN DE LAS DOS ENCUESTAS.....	103
7.4 CONCLUSIÓN FINAL DEL PRESENTE CAPÍTULO.....	106
 8. ENTREVISTA CON DOCENTES DE INGENIERÍA DE SISTEMAS Y COMPUTACIÓN DE LA NECESIDAD DE INCLUIR AL PENSUM LA METODOLOGÍA SCRUM.....	108
8.1 ANÁLISIS DE LA PRESENTE ENTREVISTA.....	113
 9. REDACCIÓN DE LA PROPUESTA DE INCLUSIÓN DE LA METODOLOGÍA SCRUM AL PENSUM DE ISC-UTP HACIENDO VER SU IMPORTANCIA PARA LOS ESTUDIANTES DEL PROGRAMA.....	114

9.1 LA ELECCIÓN DE SCRUM EN VEZ DE METODOLOGÍAS DE GESTIÓN DE PROYECTOS CONVENCIONALES PROPORCIONA MEJORES BENEFICIOS.....	114
9.2 INFORMACIÓN OBTENIDA DE LAS ENCUESTAS REALIZADAS A LOS ESTUDIANTES ACTIVOS Y EGRESADOS DE INGENIERÍA DE SISTEMAS Y COMPUTACIÓN PARA LA INCLUSIÓN DE SCRUM EN EL PROGRAMA.....	115
9.3 INFORMACIÓN OBTENIDA EN LA ENTREVISTA CON DOCENTES DE ISC-UTP AFINES A INGENIERÍA DE SOFTWARE Y LA PROMOCIÓN DE MINTIC EN EL SECTOR DE DESARROLLO DE SOFTWARE.....	117
9.4 CONCLUSIÓN DEL CAPITULO.....	118
10. CONCLUSIONES.....	119
10.1 LOS ESTUDIANTES OBTIENEN MAYORES COMPETENCIAS APRENDIENDO SCRUM SI ES INCLUIDO EN EL PROGRAMA.....	119
10.2 CONCLUSIONES DE ENCUESTAS REALIZADAS A LOS ESTUDIANTES ACTIVOS Y EGRESADOS.....	119
10.3 CONCLUSIÓN DE LA ENTREVISTA CON DOCENTES Y LA PROMOCIÓN DE MINTIC EN EL SECTOR DE DESARROLLO DE SOFTWARE.....	120
11. BIBLIOGRAFÍA.....	121
ANEXOS.....	125

LISTA DE ANEXOS

pág.

ANEXO A Encuesta de estudiantes activos de Ingeniería de Sistemas.....125

ANEXO B Encuesta de estudiantes egresados de Ingeniería de Sistemas....137

LISTA DE FIGURAS

pág.

Figura 1. Uso de metodologías.....	17
Figura 2. Metodología en Cascada.....	41
Figura 3. Metodología Ágil.....	42
Figura 4. El manifiesto.....	45
Figura 5. Estructura Scrum.....	53
Figura 6. Product Owner.....	59
Figura 7. Taskboard.....	63
Figura 8. Eventos Scrum.....	64
Figura 9. Sprint.....	65
Figura 10. Sprint planning.....	66
Figura 11. Entradas / Salidas de un Sprint Planning.....	68
Figura 12. Origen del Product Backlog.....	74
Figura 13. Sprint backlog.....	75
Figura 14. Desarrollo del Sprint backlog.....	76
Figura 15. Burn down chart.....	78
Figura 16. La jerarquía de las necesidades, según maslow.....	86
Figura 17. Certificados Professional Scrum.....	92
Figura 18. Roles de Scrum frente a las certificaciones.....	95

RESUMEN

El objetivo de este proyecto de grado es proponer la inclusión al pensum del programa de ingeniería de Sistemas y Computación de la Universidad Tecnológica de Pereira un método ágil de gestión de desarrollo de software como Scrum para con dicha metodología mejorar el trabajo en equipo en esta área en los estudiantes y lograr que al convertirse en egresados puedan obtener un mejor desempeño en el trabajo de desarrollo de software; además poder estar en la línea que viene promocionando MINTIC-Icetex [1] desde el año 2010 para logra hacer de Colombia una potencia mundial en la línea de desarrollo con altos niveles de calidad y eficiencia.

Se encontrará las características principales de Scrum y cómo optimiza el desarrollo de un proyecto de software la utilización de esta metodología que es la más difundida y utilizada en el país y el mundo (ver figura 1), para que sea dada la parte teórica en la asignatura que el programa considere mejor y después se aplique de una manera práctica en laboratorio de Software.

[1] FONDO PARA CERTIFICACIÓN EN COMPETENCIAS ESPECÍFICAS Y TRANSVERSALES (Segunda Convocatoria)

<https://www.icetex.gov.co/dnnpro5/Portals/0/Documentos/Fondos/2aConvCompetenciasTransversales15ene2015.pdf>

INTRODUCCIÓN

En los últimos años se ha vuelto muy importante el uso de metodologías ágiles para el desarrollo de software, incorporando procesos que ayuden a un excelente desempeño al trabajo colaborativo para lograr las metas establecidas, por lo tanto es importante resaltar que los ingenieros de sistemas y computación de la universidad tecnológica de Pereira, deben tener conocimiento al respecto, ya que como profesionales en el campo de la informática y la computación se hace indispensable saber optimizar procesos y resolver problemas afines, entre ellos el desarrollo de software.

Se realizó una investigación en el tema de metodologías ágiles de desarrollo de Software, profundizando en el marco de trabajo Scrum, el cual se propone su inclusión al pensum de Ingeniería de Sistemas y Computación de la Universidad Tecnológica de Pereira (la dirección del programa es quien toma la decisión si acepta o no la propuesta), esta investigación se apoyó con encuestas a estudiantes y egresados así como también con opiniones de profesores del programa de Ingeniería de Sistemas y Computación y profesionales que trabajan como directores de proyectos, utilizando la estructura de trabajo Scrum como metodología ágil en desarrollo de Software.

1. DEFINICIÓN DEL PROBLEMA

1.1 ANTECEDENTES DEL PROBLEMA

Desde los primeros semestres del programa de ingeniería de sistemas y computación de la Universidad Tecnológica de Pereira (ISC-UTP), se orientan asignaturas que enseñan desde programación básica hasta programación orientada a objetos, con diversos lenguajes de programación, entre ellos Dr. Racket al inicio con el paradigma de programación funcional, lenguaje C como paradigma imperativo, Prolog dentro del paradigma de programación Lógica, C++ y Java que pertenece al paradigma de la programación orientada a objetos (POO), también el lenguaje Python como multiparadigma (Paradigma imperativo, funcional y orientado a objetos). Previa revisión curricular por parte del programa ISC-UTP, con la debida colaboración de los diferentes estamentos que intervienen en la calidad académica del mismo.

En ISC-UTP además de las asignaturas anteriores se establecen otras como complementos a la línea de programación entre ellas: lógica matemática, bases de datos, Ingeniería de Software 1, 2 y 3, compiladores, sistemas operativos, sistemas distribuidos, inteligencia artificial, computación blanda, arquitectura cliente servidor, administración de proyectos de software y laboratorio de software, con todo este conocimiento los estudiantes de ISC-UTP adquieren herramientas para dar solución a problemas cotidianos que requieren de sistemas computacionales.

1.2 FORMULACIÓN DEL PROBLEMA

Durante la enseñanza de las diferentes asignaturas en ISC-UTP los estudiantes adquieren herramientas para dar solución a problemas cotidianos que requieren de sistemas computacionales pero en la gran mayoría de las asignaturas dichos problemas se resuelven de manera individual y por eso se forman la idea errada que todo el ciclo de vida del software lo realizan ellos mismos; solo hasta los dos últimos semestres del programa se promueve el trabajo en equipo en laboratorio de software y en la electiva A1 ingeniería de software iii.

En las asignaturas afines a ingeniería de software y la programación se orientan temas importantes para la formación de un ingeniero de Sistemas y computación pero junto a esto es muy importante desarrollar en los estudiantes el buen trabajo colaborativo en grupos que ayuden a que los problemas se solucionen de una manera organizada siguiendo una serie de parámetros que ayuden a su ejecución y culminación de una manera eficiente.

En la Universidad Tecnológica de Pereira existen varios grupos entre ellos el de las maratones de programación, que ayudan a que el estudiante aprenda a solucionar problemas computacionales acompañados de otros compañeros, también en los semilleros y grupos de investigación se realizan trabajos en equipo, pero infortunadamente a todos estos grupos llega una pequeña minoría del total de los 931 estudiantes que hay en el programa ISC-UTP.

Muchos profesionales recién graduados del programa de ISC-UTP se enfrentan en el campo laboral con metodologías de desarrollo de proyectos de software que desconocen o apenas tienen una idea muy superficial de ellas. Estas metodologías requieren de un buen trabajo en equipo ya que se necesita que cada ingeniero tenga una mayor fortaleza en la ejecución de un rol en dicho equipo, si no se tiene esta destreza, los profesionales involucrados no se

desenvuelven de una manera que le aporte al el equipo de desarrollo por no tener una comunicación óptima al momento de realizar tareas específicas que requieran de interacción mutua, debido a que en la mayoría de las asignaturas del programa se trabajó generalmente sólo al enfrentarse a dar solución a algún problema mediante un desarrollo de software.

¿Es posible aumentar el porcentaje de egresados que mejoran el trabajo en equipo; haciendo desarrollo de software con la enseñanza y aplicación de la metodología Scrum [2] en el transcurso del programa de ISC-UTP?

[2] HUNDERMARK, Peter. (2009). Un Mejor Scrum. Recuperado de <http://www.scrumsense.com>

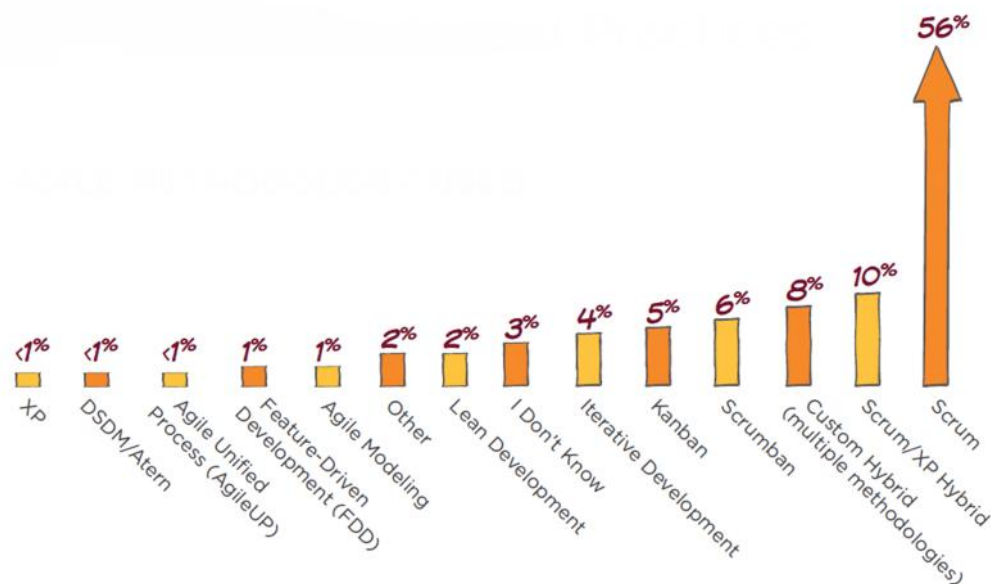
2. JUSTIFICACIÓN

En este trabajo de grado se divulgarán algunas de las características de Scrum para dimensionar todas las ventajas que aportaría a los estudiantes el incluir la metodología Scrum en el programa de Ingeniería de Sistemas y Computación de la Universidad Tecnológica de Pereira, La enseñanza de una metodología ágil como Scrum ayudará a que los estudiantes logren aprender cómo se trabaja colaborativamente en equipo en un proyecto de desarrollo de software afianzando así los métodos de trabajo que propone la metodología para una buena práctica de la misma y un desempeño ágil. Los estudiantes aprenderán a saber que en los desarrollos de proyectos de software se asignan unos roles específicos para gestionar de una manera eficiente el desarrollo y no es un trabajo de un solo ingeniero, logrando así también que los estudiantes recién graduados sepan desempeñarse de la mejor manera posible al momento de enfrentarse con una metodología ágil, que permite actuar óptimamente en equipo, trabajando en iteraciones no mayores a cuatro semanas, se logrará el aumento de personal con aptitudes de planificación, análisis de requisitos, diseño, codificación, revisión y documentación, llegando a la meta de tener un módulo de software funcional que se le presenta el cliente.

Desde el 2010 hasta la fecha de este trabajo el Ministerio de las tecnologías de la información y las comunicaciones (www.mintic.gov.co) busca hacer a Colombia una potencia mundial en desarrollo de software invirtiendo enorme cantidad de recursos para que los estudiantes y profesionales afines a las carreras de ingeniería de sistemas aprendan y se certifiquen en cualquiera de los roles que ofrece Scrum y otras metodologías complementarias como tsp/psp, con créditos condonables hasta del 100% y así lograr que todos los procesos en la industria TI se hagan de una manera más rápida y eficiente. [3]

Otro valor agregado que se logra con el aprendizaje de una metodología ágil es ponerlo en práctica en laboratorio de software y al graduarse los estudiantes de ISC-UTP van a hacer su entrada al campo laboral de una manera más acorde a la forma como se trabaja en la mayoría de la industria TI, ya que encontrarán que las metodologías ágiles son usadas ampliamente por las empresas desarrolladoras de software en Colombia y el mundo como lo podemos observar en la figura 1, donde se aprecia que Scrum se usa en el 56% y el 10% usa Scrum y XP de una manera híbrida.

Figura 1. Uso de Metodologías



Fuente: <http://info.versionone.com/state-of-agile-development-survey-ninth.html>

(Al llenar el formulario el reporte con la figura llegará al correo ingresado)

[3] ICETEX

<https://www.icetex.gov.co/dnnpro5/es->

<co/fondos/fondosparaeldesarrollodeti/certificaci%C3%B3ncomprtransversalesyespec%C3%ADficas.aspx>

Es importante resaltar que al incluir Scrum en el pensum de ISC-UTP se aumentará drásticamente el porcentaje de estudiantes que tienen una buena capacidad de trabajo en equipo como los grupos de las maratones de programación, los semilleros y grupos de investigación.

3. OBJETIVOS

3.1 OBJETIVO GENERAL

Proponer la inclusión al pensum del programa de ingeniería de Sistemas y Computación de la Universidad Tecnológica de Pereira un método ágil de gestión de desarrollo de software como Scrum para mejorar la capacidad de los estudiantes del trabajo en equipo en la línea de Ingeniería de Software.

3.2 OBJETIVOS ESPECÍFICOS

3.2.1. Realizar un análisis de las características principales de la metodología ágil Scrum para conocer su forma de trabajo, sus ciclos de desarrollo y los valores que ofrece al hacer desarrollo de proyectos de software en equipo a través de diferentes roles que funcionan de una manera organizada.

3.2.2. Consultar a una muestra representativa de estudiantes activos de sexto semestre en adelante y egresados del programa de ISC-UTP para conocer los niveles de gusto por programar, preferencias de lenguajes de programación, que metodologías de desarrollo de software conocen, cómo creen que se hace un desarrollo de software trabajado en equipo y por último saber en qué área de las TIC les gustaría trabajar después de terminar la ingeniería.

3.2.3. Entrevista con docentes del programa de Ingeniería de Sistemas y Computación que manejan áreas de programación o ingeniería de software en donde primero se les cuenta un poco la idea del proyecto y se les pide su opinión sobre la necesidad de incluir al pensum la metodología Scrum de acuerdo a lo que se ha orientado hasta el momento. También entrevistar a dos empresarios de la industria del desarrollo de software en Pereira para conocer

qué metodologías de desarrollo usan actualmente y conocer su opinión acerca del proyecto de la inclusión de la metodología Scrum en ISC-UTP.

3.2.4. Conclusión de los tres objetivos específicos anteriores que permite hacer la redacción de la propuesta de inclusión de la metodología Scrum al pensum de ISC-UTP haciendo ver su importancia para los estudiantes del programa.

4. MARCO REFERENCIAL

4.1 MARCO HISTÓRICO

En el programa de Ingeniería de Sistemas y Computación de la universidad tecnológica de Pereira se ven las siguientes diez y ocho asignaturas que preparan a los estudiantes para desarrollar software, de cincuenta y ocho asignaturas en total, lo que representa el 31.03% del pensum.

Objetivos generales de cada una de ellas: (Objetivos extraídos de la página Web de ISC-UTP citada al final de los objetivos de las asignaturas) [4].

Introducción a la Informática el objetivo es introducir algunos métodos y conceptos básicos de la matemática discreta y describir algunas de sus aplicaciones en las ciencias de la computación.

Programación I el objetivo es formar en el estudiante las competencias necesarias para formular soluciones a problemas computacionales soportado en el paradigma de programación funcional.

Programación II el objetivo es para un problema algorítmico de mediana y alta complejidad el estudiante diseñe un algoritmo, mediante un enfoque funcional elemental, que lo solucione y por el otro implementarlo en un lenguaje de programación imperativa.

Estructura de datos el objetivo es construir a partir de los fundamentos básicos, diferentes tipos Abstractos de datos que modelen distintas estructuras de datos para la solución de problemas computacionales que se enmarquen en el dominio de estas estructuras específicas.

[4] INGENIERÍA DE SISTEMAS Y COMPUTACIÓN, isc.utp.edu.co/informacion/pensum-ingenieria-de-sistemas-y-computacion-.html

Lógica el objetivo es brindar al estudiante la fundamentación teórica necesaria para entender el cálculo proposicional y de predicados, para obtener así una estructura teórica lógica correcta, que le permita fundamentar su accionar en el resto de su carrera.

Programación III el objetivo es conocer los fundamentos de la programación lógica, resolver problemas por medio de la programación lógica, Aplicar técnicas de programación con restricciones, Aplicar las técnicas de la programación lógica en un lenguaje.

Programación IV el objetivo es estudiar los fundamentos del paradigma de programación orientado a objetos e igualmente aplicarlo a la solución de problemas de la vida real.

Bases de Datos I el objetivo es proporcionar a los futuros ingenieros, fundamentos teóricos y prácticos en el diseño de bases de datos y los sistemas de gestión de bases de datos (SGBD).

Ingeniería de Software I el objetivo es presentar los temas necesarios para que al terminar el curso el estudiante desarrolle las competencias propias de la disciplina del desarrollo de software, independientemente de las herramientas tecnológicas disponibles en el mercado.

Compiladores el objetivo es al finalizar el curso, el estudiante estará en capacidad de construir un compilador para un lenguaje específico de mediana complejidad. El compilador construido tendrá incorporados los módulos para el análisis lexicográfico, análisis sintáctico, análisis semántico, generación de código intermedio y generación de código ejecutable (*.EXE y *.DLL).

Sistemas Operativos I el objetivo es entregar al estudiante conceptos básicos de diseño e implementación de sistemas operativos en el modelo centralizado. Al finalizar el curso, el alumno entenderá los problemas que debe atender un sistema operativo, como administrador de recursos y podrá abordar y conocer diversos mecanismos para resolverlos.

Asimismo, estará capacitado para diseñar e implementar rutinas que formen parte de un sistema operativo en este esquema.

Inteligencia Artificial el objetivo es dar una visión general, tanto en el aspecto teórico como en el algorítmico y en las aplicaciones, de los principales problemas que se trabajan en el área de la Inteligencia Artificial.

Sistemas Distribuidos el objetivo es dar una visión general, tanto en el aspecto teórico como en el algorítmico y en las aplicaciones, de los principales problemas que se trabajan en el área de la Inteligencia Artificial.

Ingeniería de Software II el objetivo es presentar técnicas y herramientas propias de la ingeniería de software que le garanticen al estudiante mantener el nivel competitivo adecuado al mercado del desarrollo de software, de acuerdo con las tendencias internacionales.

Computación Blanda el objetivo es profundizar, tanto en el aspecto teórico como en el algorítmico y en las aplicaciones, en los métodos en IA de inspiración biológica.

Arquitectura Cliente Servidor el objetivo es desarrollar la capacidad de implementación de sistemas C/S de alto rendimiento, y conocer sus características y posibilidades de aplicación en diferentes áreas de los Sistemas Distribuidos.

Laboratorio de Software el objetivo es utilizar los conocimientos teóricos y prácticos adquiridos durante la formación profesional para resolver un problema del entorno, aplicando la ingeniería de software y llevando a cabo las fases que conforman el ciclo de vida del software.

Administración de Proyectos de Software el objetivo es conocer los conceptos y los elementos básicos de un proyecto de software, así como los factores críticos de éxito; empleando metodologías, técnicas y herramientas de gestión de proyectos específicos de software.

Ingeniería de Software III: Brindar a los estudiantes herramientas para facilitar el uso de patrones, metodologías y propuestas arquitectónicas en el proceso de desarrollo de software, con el propósito de aumentar la calidad de las aplicaciones que se construyen tanto en el ámbito académico como en el profesional.

Al enunciar todos los objetivos de las asignaturas que tienen que ver con desarrollo de software se ofrece a la dirección de ISC-UTP una visión general del contenido que está planteado a la fecha de realización de este proyecto de grado en dicho programa, en estos objetivos se puede apreciar que en ninguno de ellos aparece el estudio y/o aplicación de la metodología Scrum lo que nos indica que si se acepta la inclusión de enseñar esta metodología se incide positivamente a los futuros ingenieros de sistemas y computación.

Con respecto a Scrum en una reunión celebrada en febrero de 2001 en Utah (EEUU), nace el término “ágil” aplicado al desarrollo de software. En esta reunión participaron un grupo de 17 expertos de la industria del software. Su objetivo fue esbozar los valores y principios que debían permitir a los equipos desarrollar software rápidamente y respondiendo a los cambios que pueden surgir a lo largo del proyecto. Se pretendía ofrecer una alternativa a los procesos de desarrollo de software tradicionales, caracterizados por ser rígidos y dirigidos por la documentación que se genera en cada una de las actividades desarrolladas. [5]

Tras esta reunión se creó The Alliance, una organización dedicada a promover los conceptos relacionados con el desarrollo ágil de software y ayudar a las organizaciones para que los adopten. El punto de partida fue el Manifiesto Ágil, un documento que resume la filosofía ágil.

[5] Metodologías ágiles para el desarrollo de software, <http://www.cyta.com.ar/ta0502/v5n2a1.htm>

4.2 MARCO CONCEPTUAL

4.2.1 Metodología Ágil: The Agile Alliance, [6] es una organización, sin ánimo de lucro, dedicada a promover los conceptos relacionados con el desarrollo ágil de software y fue creada en el 2001.

Durante el transcurso de los años 90 el ambiente cambiante y turbulento era cada vez más la regla que la excepción. Por lo tanto surgió la necesidad de desarrollar metodologías livianas y maniobrables que pudieran operar en ese ambiente turbulento. Estas metodologías son conocidas colectivamente hoy en día como “metodologías ágiles”.

Lo ágil se define (por los mismos agilistas) como la habilidad de responder de forma versátil al cambio para maximizar los beneficios. Las metodologías ágiles varían en su forma de responder al cambio, pero en general comparten las siguientes características:

- Los individuos y sus interacciones son más importantes que los procesos y las herramientas.

- El software que funciona es más importante que la documentación exhaustiva.

- La colaboración con el cliente en lugar de la negociación de contratos.

- La respuesta al cambio en lugar de aferrarse a un plan.

Los valores y principios compartidos por toda la metodología ágil fueron enunciados en el “manifiesto ágil”, por la “alianza ágil”.

[6] Metodología ágil de desarrollo de software extremo

<http://www.monografias.com/trabajos67/metodologia-desarrollo-sofware/metodologia-desarrollo-sofware2.shtml>

4.2.2 Metodología Ágil Scrum: Scrum nació en 1980 como nuevas prácticas de producción de productos tecnológicos en Japón y se aplicó al desarrollo de software en 1993. Scrum es el más popular de las metodologías ágiles con un 56% y se utiliza frecuentemente en conjunto con Extreme Programming (XP), llevando así al Scrum Master como el mejor conocedor o el mejor experto con un 44% en metodologías ágiles. “Scrum provee una plataforma para que la gente trabaje en conjunto de manera efectiva, a la vez que permite exponer de manera implacable cualquier problema que se ponga en su camino” [7]

El alto grado de proyectos fallidos e insatisfacción de los clientes ilustra con claridad de los problemas que surgen al momento de no tener como base una metodología para llevar a cabo el desarrollo de un proyecto de software ya que los ciclos de retroalimentación son demasiado largos o incluso no existen.

Si no se tiene una metodología es deficiente la gestión y el monitoreo de proyectos de software, por lo que se plantea la implementación de la metodología Scrum.

[7] HUNDERMARK, Peter. (2009). Un Mejor Scrum. Recuperado de <http://www.scrumsense.com>

4.3 MARCO TEÓRICO

El manifiesto ágil.

El Manifiesto comienza enumerando los principales valores del desarrollo ágil.

Se valora:

Al individuo y a las interacciones del equipo de desarrollo sobre el proceso y las herramientas. La gente es el principal factor de éxito de un proyecto de software. Si se sigue un buen proceso de desarrollo, pero el equipo falla, el éxito no está asegurado; sin embargo, si el equipo funciona, es más fácil conseguir el objetivo final, aunque no se tenga un proceso bien definido. Así mismo, las herramientas (compiladores, depuradores, control de versiones) son importantes para mejorar el rendimiento del equipo, pero el disponer de más recursos de los estrictamente necesarios también puede afectar negativamente.

Desarrollar software que funciona más que conseguir una buena documentación. Aunque se parte de la base de que el software sin documentación es un desastre, la regla a seguir es “no producir documentos a menos que sean necesarios de forma inmediata para tomar una decisión importante”. Estos documentos deben ser cortos y centrarse en lo fundamental. Si una vez iniciado el proyecto, un nuevo miembro se incorpora al equipo de desarrollo, se considera que los dos elementos que más le van a servir para ponerse al día son: el propio código y la interacción con el equipo.

La colaboración con el cliente más que la negociación de un contrato. Las características particulares del desarrollo de software hacen que muchos proyectos han fracasado por intentar cumplir plazos y costes preestablecidos al inicio del mismo, según los requisitos que el cliente manifestaba en ese momento. Por ello, se propone que exista una interacción constante entre el cliente y el equipo de desarrollo. Esta colaboración entre ambos será la que marque la marcha del proyecto y asegure su éxito.

Responder a los cambios más que seguir estrictamente un plan. La habilidad de responder a los cambios que puedan surgir a lo largo del proyecto (en los requisitos, la tecnología, el equipo) determina también el éxito o fracaso del mismo. Por lo tanto, la planificación debe ser flexible para poder adaptarse a los cambios que puedan surgir. Una buena estrategia es hacer planificaciones detalladas para unas pocas semanas y planificaciones mucho más abiertas para unos pocos meses.

Los valores anteriores inspiran los doce principios del manifiesto de los cuales se hablará en el capítulo 6. Éstas son las características que diferencian un proceso ágil de uno tradicional. Los dos primeros son generales y resumen gran parte del espíritu ágil.

4.4 MARCO GEOGRÁFICO

La propuesta de inclusión de la metodología Scrum se realizará para el programa de Ingeniería de Sistemas y Computación de la Universidad Tecnológica de Pereira con coordenadas: 4°47'35.9"N 75°41'23.0"W 4.793316, -75.689717

5. DISEÑO METODOLÓGICO

5.1 UNIVERSO O POBLACIÓN

Estudiantes activos: Los 679 estudiantes de la jornada diurna y 252 estudiantes de la jornada nocturna para un total de 931 estudiantes del programa de Ingeniería de Sistemas y Computación de la Universidad Tecnológica de Pereira (número de estudiantes del segundo semestre de 2014)

Egresados: 250 desde 2010 al 2014 de Ingeniería de Sistemas y Computación.

Docentes: 40 del programa que tienen asignaturas afines a la propuesta.

5.2 MUESTRA

El cálculo de la muestra de estudiantes activos se realizó de la siguiente manera: [8]

TAMAÑO DEL UNIVERSO: Número de personas que componen la población a estudiar: **358**

MARGEN DE ERROR: **14%**

HETEROGENEIDAD: Es la diversidad del universo: **50%** (Lo habitual suele ser 50%)

NIVEL DE CONFIANZA: **95%** (Lo habitual suele ser entre el 95% y el 99%)

TAMAÑO DE LA MUESTRA RECOMENDADO: **44**

[8] NETQUEST

<http://www.netquest.com/es/panel/calculadora-muestras.html>

La muestra que se va a tomar para la validación del objetivo específico 3.2.2 será de 44 estudiantes activos desde quinto semestre hasta último semestre lo que equivale al 4.7% del total del universo.

El cálculo de la muestra de egresados se realizó de la siguiente manera:

TAMAÑO DEL UNIVERSO: Número de personas que componen la población a estudiar: **250**

MARGEN DE ERROR: **32%**

HETEROGENEIDAD: Es la diversidad del universo: **50%** (Lo habitual suele ser 50%)

NIVEL DE CONFIANZA: **95%** (Lo habitual suele ser entre el 95% y el 99%)

TAMAÑO DE LA MUESTRA RECOMENDADO: **10**

La muestra que se va a tomar para la validación del objetivo específico 3.2.2. será de 10 egresados lo que equivale al 4% del total del universo.

El cálculo de la muestra de docentes se realizó de la siguiente manera:

TAMAÑO DEL UNIVERSO: Número de personas que componen la población a estudiar: **40**

MARGEN DE ERROR: **30%**

HETEROGENEIDAD: Es la diversidad del universo: **50%** (Lo habitual suele ser 50%)

NIVEL DE CONFIANZA: **95%** (Lo habitual suele ser entre el 95% y el 99%)

TAMAÑO DE LA MUESTRA RECOMENDADO: **9**

La muestra para la validación del objetivo específico 3.2.3. será de 9 docentes lo que equivale al 22.5% del total del universo y se complementará con dos profesionales en administración de proyectos de software que trabajan actualmente en empresas desarrolladoras de software.

Los docentes serán distribuidos de la siguiente manera:

- 1 docente de programación 3. (Alejandro Rodas)
- 1 docente de programación 4. (Alejandro Rodas)
- 1 docente de Ingeniería del software 1.(Jorge Gálvez)
- 1 docente de Ingeniería del software 3.(Carlos Ocampo)
- 1 directora del grupo de investigación Grande. (Luz Stella Valencia)
- 1 docente de sistemas operativos 1. (Juan Veloza)
- 1 docente arquitectura cliente / servidor. (Carlos M Medina)
- 1 docente de laboratorio de software. (Wilman Tello)
- 1 docente de administración de proyectos de software. (Guillermo González)

5.3 DEFINICIÓN DE HIPÓTESIS

La inclusión de una metodología ágil como Scrum en el pensum del programa de Ingeniería de Sistemas y Computación de la Universidad Tecnológica de Pereira logrará que los estudiantes alcancen el objetivo de mejorar el desempeño para trabajar en equipo en desarrollo de proyectos de software para tener claro cómo se trabaja en la industria del sector donde hay diferentes personas que cumplen un rol particular para lograr así mayor eficiencia y entender que todos estos procesos no son tarea de un solo ingeniero.

5.4 VARIABLES E INDICADORES

5.4.1 Definir líneas de profundización en los estudiantes.

5.4.2 Eficiencia en desarrollo de software

5.4.3 Desempeño en desarrollo de software

5.4.4 Control en desarrollo de software

5.5 INSTRUMENTOS DE RECOLECCIÓN DE INFORMACIÓN

Se realizó encuestas con formularios virtuales para consultar a estudiantes activos y egresados con diferentes preguntas que se pueden ver respondidas con valores y gráficos (ver anexo A y anexo B), también se entrevistó a nueve docentes que manejan áreas de programación o ingeniería de software y dos empresarios de Pereira de la industria en desarrollo de Software para saber su opinión sobre el presente proyecto para proponer incluir Scrum en ISC-UTP (ver capítulo 8 con toda la información de las entrevistas a docentes y empresarios).

5.6 PLAN DE ANÁLISIS DEL ESTUDIO PILOTO

Según Netquest [9] el plan de análisis se hará todo con tablas estadísticas con parámetros estadísticos universales y con fichas técnicas, se explicará en detalle cada una de las dos encuestas con una conclusión al final y el análisis de los datos obtenidos

5.6.1 Encuesta que se realizó a los estudiantes activos de sexto semestre al último semestre de la Universidad Tecnológica de Pereira del programa de Ingeniería de Sistemas y Computación y con las respuestas se calculó:

Porcentajes de cuantos semestres le faltan para culminar sus estudios, para así saber qué porcentaje están en último semestre y que porcentaje están en los semestres anteriores.

Porcentajes de estudiantes que pertenecen a cada jornada.

Porcentaje de preferencia de lenguajes de programación seleccionando entre diez opciones de respuesta (Pregunta de respuesta múltiple, lo que hará que la suma de porcentajes no de 100%).

Porcentaje de gusto por la programación de los estudiantes.

Porcentaje de entre cinco opciones de respuesta que va entre muy bien hasta muy mal, para evaluar cómo le fue en las asignaturas de programación.

Pregunta abierta, que después será agrupada para cálculo porcentual de cómo cree que se hace un desarrollo de software en equipo para saber el porcentaje de quienes lo harían con métodos ágiles y que porcentaje no sabe la forma adecuada de hacerlo.

Porcentaje en SI y NO si alguna vez ha escuchado sobre metodologías ágiles para desarrollo de Software.

[9] NETQUEST, <http://www.netquest.com/es/panel/calculadora-muestras.html>

Porcentaje de cuatro opciones de respuesta para calcular el nivel de conocimiento de la metodología Scrum, donde se logró obtener el porcentaje de estudiantes que conocen perfectamente la metodología Scrum y quienes la conocen poco.

En qué área del conocimiento le gustaría ejercer su profesión de diez propuestas para luego calcular el porcentaje de cada una. (Pregunta de respuesta múltiple, lo que hará que la suma de porcentajes no de 100%)

Con todos los resultados porcentuales de las preguntas realizadas, se logró un análisis de la importancia de incluir la metodología Scrum al pensum de ISC-UTP (Todos los valores porcentuales y el análisis detallado se encuentra en el capítulo 7 del presente trabajo)

5.6.2 Encuesta que se realizó a los estudiantes egresados de la Universidad Tecnológica de Pereira del programa de Ingeniería de Sistemas y Computación, con las respuestas se calculó:

Porcentaje respecto al año de graduación donde la muestra tomada indica que los egresados se graduaron entre los años 2010 y 2014 y se calculó el porcentaje de alumnos graduados en cada uno de estos años.

Porcentaje de lenguajes de programación preferidos escogiendo entre diez opciones de respuesta (Pregunta de respuesta múltiple, lo que hará que la suma de porcentajes no de 100%).

Porcentaje de evaluación de cómo le fue en las asignaturas de programación escogiendo entre cinco opciones de respuesta que va entre muy bien hasta muy mal.

Porcentaje del gusto a programar con las opciones de Si y No.

Porcentaje de entre cinco opciones de respuesta que va entre muy bien hasta muy mal, para evaluar cómo le fue en las asignaturas de programación.

Porcentaje de las respuestas a la pregunta si trabaja o a trabajado en desarrollo de Software utilizando alguna metodología que le hayan enseñado en la Universidad en su formación, donde se obtuvo el porcentaje de egresados que aprendió en las asignaturas y el porcentaje de egresados que aprendió en el trabajo (Pregunta abierta, que después será agrupada para cálculo porcentual).

Pregunta abierta, que después será agrupada para cálculo porcentual de cómo cree que se hace un desarrollo de software en equipo para saber el porcentaje de quienes lo harían con métodos ágiles, que porcentaje con PSP y que porcentaje no sabe la forma adecuada de hacerlo.

Porcentaje de respuestas a la pregunta si alguna vez a escuchado sobre metodologías ágiles para desarrollo de Software con opción de respuesta de Si y NO.

Porcentaje de las respuestas a la pregunta si conoce la metodología Scrum ofreciendo cuatro opciones de respuesta para poder calcular el nivel de

conocimiento al respecto.

Porcentaje del área del conocimiento en el que le gustaría ejercer su profesión pudiendo escoger entre varias opciones de respuesta de diez propuestas (Pregunta de respuesta múltiple, lo que hará que la suma de porcentajes no de 100%)

Con todos los resultados porcentuales de las preguntas realizadas, se logró un análisis de la importancia de incluir la metodología Scrum al pensum de ISC-UTP (Todos los valores porcentuales y el análisis detallado se encuentra en el capítulo 7 del presente trabajo)

6. ANÁLISIS DE LAS CARACTERÍSTICAS PRINCIPALES DE LA METODOLOGÍA SCRUM PARA CONOCER SU FORMA DE TRABAJO, SUS CICLOS DE DESARROLLO Y LOS VALORES QUE OFRECE AL HACER DESARROLLO DE PROYECTOS DE SOFTWARE EN EQUIPO A TRAVÉS DE DIFERENTES ROLES.

Según el ministerio de tecnologías de la información y las Comunicaciones (MINTIC) [10] desde el 2010, lanzó un plan para consolidar a la industria TI en Colombia de clase mundial apoyando el fortalecimiento de la industria TI y el talento digital, primero fue el plan vive digital 2010-2014, actualmente está en el plan vive digital 2014-2018.

En el 2014 el total de empresas TI fueron 1.850 y se espera crecer en el 2018 a 3.700.

El número de empresas certificadas en calidad en el 2014 fue de 120 y se espera tener 360 empresas certificadas en el 2018.

El número de empleados TI en el 2014 fue de 41.000 en el 2018 se proyecta en 369.000.

En 2010 el total de la industria de TI en Colombia facturó 3.28 billones de pesos en el 2014 paso a facturar 6 billones de pesos y se espera lograr que en el 2018 la facturación en TI sea de 18 billones de pesos.

[10] MINTIC, <http://www.mintic.gov.co/portal/604/w3-channel.html>

Entre las múltiples estrategias que implementa MINTIC para fortalecer la industria TI y lograr los valores propuestos esta la capacitación a personal TI con créditos del ICETEX 100% condonables por MINTIC, uno de ellos es la certificación en competencias transversales que consta de las siguientes cursos: Scrum master, Product Owner, Certified Negotiation Expert Professional, Project management Professional (PMP) con énfasis en: Desarrollo de Software, Information Technology y en Project Management, Leadership and Communication (PMLC), Establishing and Managing the Project Management Office PMO, TI.+ Leadership And Management Skills) (LMS). Al igual que las certificaciones en: ITIL y 9 ISO/IEC 20000:2011 Information Technology Service Management.

“Como vemos los dos primeros son certificación en Scrum master y Product Owner” [11], dos roles que se explicara detalladamente en el presente capítulo.

Segun el Instituto PMI [12] en la actualidad muchas organizaciones que se dedican al desarrollo de proyectos de Software se encuentran en el dilema de considerar un cambio para acelerar sus procesos de desarrollo y no están seguros de qué forma realizar este cambio sin introducir riesgos en la calidad de software o en el cumplimiento de políticas y normas organizacionales.

[11] INTERSOFTWARE, Certifíquese en SCRUM con un crédito 100% condonable, <http://www.intersoftware.org.co/content/certifiquese-en-scrum-con-un-credito-100-condonable>

[12] PMI, Project Manage Institute, <http://americalatina.pmi.org/latam/AboutUS/WhatIsPMI.aspx>

6.1 PORQUÉ UTILIZAR METODOLOGÍAS DE GESTIÓN DE PROYECTO

Un proyecto tiene una definición muy específica, según el Instituto de Gestión de Proyectos (Project Management Institute - PMI), un proyecto se compone de un conjunto de actividades temporales destinadas a producir un resultado específico. Para calificar como un proyecto, estas actividades deben llevarse a cabo dentro de un marco de tiempo de inicio y final definido.

Cuanto más complejo sea el proyecto, mayor es la necesidad de seguir y probar metodologías para iniciar, planear, ejecutar, monitorear, controlar y finalizar el proyecto. Por lo tanto, una metodología de gestión de proyectos no es más que un conjunto de directrices probadas que ayudarán a los directivos a terminar sus proyectos a tiempo, según las especificaciones y dentro del presupuesto. El uso de una metodología formal es lo que da a los directores de proyectos la capacidad de terminar con éxito. La alternativa de no utilizar una metodología de gestión de proyectos bien definido es el fracaso garantizado.

6.2 PRESENTACIÓN DE DOS PRIMOS: CASCADA Y ÁGIL

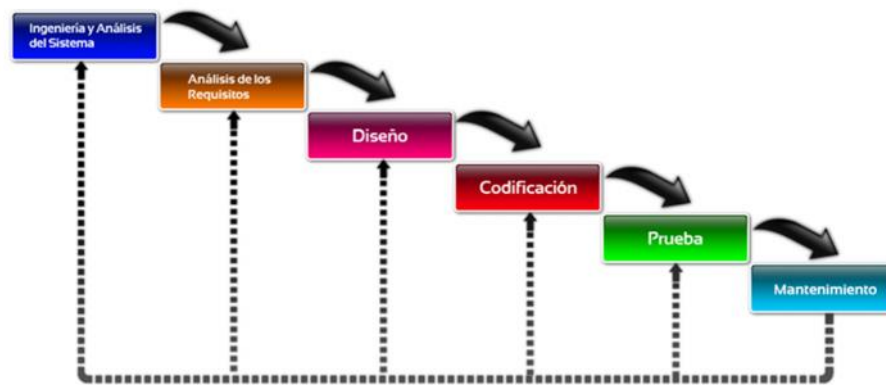
Dos metodologías de gestión de proyectos comunes son cascada y ágil, ambas metodologías se han utilizado ampliamente en los entornos de gestión de proyectos de ingeniería de software.

6.2.1 Cascada: La metodología de cascada se remonta a finales de la Segunda Guerra Mundial, y le debe su origen a la gestión de proyectos de fabricación a gran escala de la industria de la defensa. La metodología básica es el proceso guiado y proporciona a los responsables del proyecto una estructura de etapas antes que dicho proyecto se considere terminado.

Las etapas incluyen definición de los requisitos, diseño, desarrollo, pruebas y despliegue.

Una característica distintiva de la cascada es que es una metodología basada en la secuencia, donde el proyecto no puede continuar a la etapa de diseño a menos que la etapa de definición de requisito haya sido terminada exitosamente.[13].

Figura 2. Metodología en Cascada



Fuente: www.gavide.com

[13] Jefferson Hanley (2015), Scrum: Your Quick Start Guide To Adopting Scrum For Your Organization

6.2.2 Ágil: La experimentación temprana con una metodología alternativa, conocida como Ágil, tuvo lugar en la década de 1990. Mientras Cascada estipuló una serie de etapas, metodologías ágiles receta un conjunto de interacciones y directrices que los directores de proyectos se adaptan a sus necesidades de proyectos únicos.

El sello distintivo de metodologías ágiles de los cuales Scrum es uno de ellos, es que se basa en un conjunto de ciclos iterativos para completar los proyectos en pequeños conjuntos de resultados finales. Ágil pone un alto grado de énfasis en equipos autogestionados sin imponer una rígida gestión de proyectos.

El resultado final es un enfoque bastante flexible como para la definición de los requisitos del proyecto y a continuación, la entrega de una fracción del proyecto de acuerdo a las necesidades de los clientes.

Figura 3. Metodología Ágil



Fuente: Metodología Ágil Crystal Clear
<https://iswugaps2crystalclear.wordpress.com/>

6.3 SIMILARES PERO CON DIFERENCIAS

Como bien lo dice Hanley [14] las metodologías Cascada y Ágil han sido probadas a lo largo de los años y han tenido tanto defensores como detractores y se resalta que todas las metodologías tendrán proyectos exitosos como proyectos fallidos. Mientras Cascada sigue más un enfoque de adaptación, que define un conjunto de etapas distintas que se adaptan a un proyecto; Ágil es una metodología prescriptiva, en la que se establece reglas que pueden ser luego adoptadas por el director del proyecto.

Otra diferencia entre los dos está en cómo se definen las funciones y las interacciones del equipo en el desarrollo del proyecto. Los equipos Cascada son mucho más rígidos por resistirse a los cambios, usan límites jerárquicos alrededor de sus funciones mientras que los equipos ágiles son más flexibles en sus roles y se adaptan al cambio.

Además, el enfoque de la aproximación de la metodología en cascada es entregar un producto final solo hasta cuando el proyecto concluye y no realiza retrospectiva, mientras que las metodologías ágiles creen en la presentación de un producto utilizable en toda la vida del proyecto, aunque sea en pequeños incrementos realizando retrospectiva al final de cada uno. Para los proyectos de cascada, la prueba de fuego de un producto o servicio no puede llevarse a cabo hasta que todos los componentes se entregan al final del proyecto. Si hay deficiencias importantes, es necesario regresar al inicio.

[14] Jefferson Hanley (2015), Scrum: Your Quick Start Guide To Adopting Scrum For Your Organization

Esta última peculiaridad de la metodología en cascada está causando en profesionales de Dirección de Proyectos un aspecto más favorable a la metodología Ágil como una estructura de trabajo viable que aborde la cuestión en cuanto a soluciones de desarrollo que deben ser entregados por los proyectos en iteraciones.

El concepto de fracasar rápidamente, que está expuesta por la metodología Ágil, establece que mediante la entrega de proyectos en incrementos graduales, uno puede descubrir mucho antes en el cronograma del proyecto, si hay algo que no va a funcionar. Si hay un problema se descubre y se arregla fácilmente.

Ser ágil fija rápidamente los problemas y la entrega de una versión utilizable, antes de pasar a la construcción del siguiente componente, todas estas características están en el corazón de las metodologías ágiles y Scrum es la más destacada de estas metodologías.

A causa de estos métodos ágiles que estaban surgiendo, se definieron los 4 postulados que dan origen a:

Figura 4. El manifiesto



Fuente: Manuel Trigas Gallego, El manifiesto, metodología Scrum. pag. 27

Principios del Manifiesto Ágil.

Nuestra mayor prioridad es satisfacer al cliente mediante la entrega temprana y continua de software con valor.

Aceptamos que los requisitos cambien, incluso en etapas tardías del desarrollo. Los procesos Ágiles aprovechan el cambio para proporcionar ventaja competitiva al cliente.

Los responsables de negocio y los desarrolladores trabajamos juntos de forma cotidiana durante todo el proyecto.

Los proyectos se desarrollan en torno a individuos motivados. Hay que darles el entorno y el apoyo que necesitan y confiarles la ejecución del trabajo.

El método más eficiente y efectivo de comunicar información al equipo de desarrollo y entre sus miembros es la conversación cara a cara.

El software funcionando es la medida principal de progreso.

Los procesos Ágiles promueven el desarrollo sostenible. Los promotores, desarrolladores y usuarios debemos ser capaces de mantener un ritmo constante de forma indefinida.

La atención continua a la excelencia técnica y al buen diseño mejora la Agilidad.

La simplicidad, o el arte de maximizar la cantidad de trabajo no realizado, es esencial.

Las mejores arquitecturas, requisitos y diseños emergen de equipo auto organizados.

A intervalos regulares el equipo reflexiona sobre cómo ser más efectivo para a continuación ajustar y perfeccionar su comportamiento en consecuencia.

Entregamos software funcionando frecuentemente, entre un par de semanas a un par de meses, prefiriendo los tiempos más cortos.[15]

[15] HIGHSMITH Jim 2001, The Agile Manifesto

<http://www.drdobbs.com/open-source/the-agile-manifesto/184414755>

6.4 SCRUM

Scrum [16] es un proceso en el que se aplica de manera regular un conjunto de buenas prácticas para trabajar colaborativamente en equipo y obtener el mejor resultado posible de un proyecto. Estas prácticas se apoyan unas a otras y su selección tiene origen en un estudio de la manera de trabajar de equipos altamente productivos.

En Scrum se realizan entregas parciales y regulares del producto final, priorizadas por el beneficio que aportan al receptor del proyecto. Por ello, Scrum está especialmente indicado para proyectos en entornos complejos donde se necesita obtener resultados pronto, donde los requisitos son cambiantes o poco definidos, donde la innovación, la competitividad, la flexibilidad y la productividad son fundamentales.

[16] DEEMER, Pete., Benefield, Gabrielle., Larman, Craig y Vodde, Bass. (2012). A Lightweight Guide to the Theory and Practice of Scrum. Recuperado de <http://scrumfoundation.com/library>

6.4.1 La estructura de trabajo Ágil: Con el fin de ser una herramienta eficaz para sus practicantes, cualquier cuerpo de conocimiento necesita una estructura formal dentro del cual pueda funcionar. Ágil no es una excepción. Las metodologías ágiles como Scrum y otros como el sistema dinámico de Metodología de Desarrollo (DSDM) o Extreme Programming (XP), trabajan dentro de una estructura que regula cómo deben aplicarse los diferentes integrantes del equipo ágil de conocimiento para la gestión de proyectos.

A diferencia de la metodología en cascada que estipula que el proyecto se ejecute como etapas distintas con entregables que se produce al final del proyecto, la metodología Ágil prescribe un conjunto de convenciones que los gerentes de proyectos puedan adaptarse a la entrega de proyectos. Los distintivos de la estructura Agile incluyen:

Vigilancia Ágil: Que hace hincapié en la necesidad de construir circuitos de retroalimentación dentro del proyecto que continuamente monitorea y rastrea todos los aspectos del proyecto para adaptarse a este mundo de proyectos dinámicos.

Información Abierta: Esto permite el libre flujo de información y opiniones entre todas las partes interesadas en el proyecto. Esto permite que el equipo evalúe las condiciones cambiantes y rápidamente adaptar sus respuestas.

Visión guiada: El objetivo es facilitar los cambios de comportamiento que en última instancia resultan en grupos de interés internos y externos que trabajan para lograr un objetivo común. La visión es compartida por todos los que influyen en los resultados positivos lo que conduce al éxito del proyecto.

Colaboración y Trabajo en Equipo: Esta parte del estructura está destinado a promover la sana interacción entre los equipos y fomentar la cooperación entre todos con el fin de cumplir con los objetivos deseados.

Reglas Simples: La estructura Ágil reconoce que un simple conjunto de principios de reglas pueden ser utilizados para apoyar ambientes extremadamente complejos, incluyendo las interacciones y relaciones entre los miembros del equipo. Cuanto más complejas sean las reglas, mayores serán las posibilidades de tener malentendidos que llevan al fracaso el proyecto. [17]

6.4.2 ¿Qué es Scrum? En el desarrollo de la metodología Scrum, “los expertos se dieron cuenta de que tenía que haber una cierta disciplina detrás de la ciencia de la gestión de proyectos” [18].

[17] [18] Jefferson Hanley (2015), Scrum: Your Quick Start Guide To Adopting Scrum For Your Organization

Los proyectos se pueden lograr entregar a tiempo y dentro del presupuesto, cuando los directores de proyectos cumplen los siguientes parámetros:

Organizan el negocio en pequeños equipos autónomos multifuncionales.

Organizan el trabajo en pequeñas partes realizables.

Priorizan y completan la estimación y la entrega.

Organizan la entrega de pequeños componentes de trabajo en un corto plazo de tiempo (1 a 4 semanas) en Sprints (Iteraciones).

Consultan con los clientes / usuarios finales y organizar planes de entrega basado en resultados de la inspección de cada iteración.

Optimizan todo el proceso basado en la revisión retrospectiva después de cada iteración. [19]

Esta es la idea central de la que evolucionó Scrum.

A medida que el mundo de los negocios se volvió más competitivo, según Hanley [20] los promotores del proyecto estaban buscando una manera de acelerar la ejecución de los proyectos para que pudieran obtener una ventaja competitiva más rápidamente al hacer lanzamiento al mercado de su producto. Especialistas en gestión de proyectos llegaron a soportar ese objetivo con la estructura de Scrum.

[19] [20] Jefferson Hanley (2015), Scrum: Your Quick Start Guide To Adopting Scrum For Your Organization

6.4.3 La estructura Scrum: Como bien lo dice Hanley [21] entregar un proyecto conforme con los requisitos de diversas partes interesadas (Equipo Scrum y Clientes) es una tarea desafiante. Sin embargo Scrum ha facilitado este desafío investigando la prescripción de una estructura para llevar a cabo el proceso de gestión del proyecto.

Los proyectos complejos pueden ser entregados a través de la colaboración y el trabajo eficaz en equipo.

El foco principal del éxito de Scrum es tratar de gestionar con eficacia un Product Backlog (Pila de producto) de entregables que representan el todo. Esto a su vez terminó por romper el todo en pequeñas entregas conocidos como Sprint Backlog (Pila del sprint). A medida que se completa cada Sprint, añade valor a las prestaciones de los Sprints anteriores por lo que el sprint final marca la culminación del proyecto.

Dentro de esta estructura, Scrum utiliza equipos que se rigen por roles predeterminados, Eventos, artefactos y Reglas.

[21] Jefferson Hanley (2015), Scrum: Your Quick Start Guide To Adopting Scrum For Your Organization

Figura 5. Estructura Scrum



Fuente:

http://projectmanager.org/content.php?page=Certified_Scrum_Product_Owner

6.4.3.1 El equipo Scrum: Según Hanley [22] en el mundo real los equipos Scrum son un nombre colectivo dado a varios conjuntos de funciones que son realizadas por individuos y grupos de trabajo en un proyecto. La clave para el éxito del proyecto Scrum es asegurar que se forma el equipo adecuado y que todos en el equipo cumplan los siguientes requisitos:

[22] Jefferson Hanley (2015), Scrum: Your Quick Start Guide To Adopting Scrum For Your Organization.

- Están de acuerdo en perseguir los objetivos comunes.
- Abarcan un conjunto común de reglas y normas.
- Respetan todos y cada uno de los miembros del equipo.
- Valoran toda la información dada por cada miembro.

Sin estos requisitos previos en su lugar, un proyecto Scrum está condenado al fracaso desde el principio del proyecto.

Los directores de proyectos Scrum deben planificar con anterioridad de una manera flexible sus planes de formación, normalización, desempeño y suspensión de un Sprint antes que el equipo Scrum inicie su trabajo.

Los equipos Scrum son por definición, multifuncionales en su configuración. Eso significa que es probable que tenga miembros en el equipo con diversos conjuntos de habilidades, muchos de los que nunca han trabajado juntos y algunos que probablemente serán parte de un equipo de muy corta duración entonces serán reemplazados por otros.

6.4.3.2 Roles de Scrum: El Equipo Scrum debe ser organizado alrededor de unos roles específicos de trabajo.

Scrum Master: Según lo dice Hanley [23] esta función asegura que el equipo siga las mejores prácticas de Scrum y se adhieran a las normas y reglas que el equipo se compromete a seguir. Normalmente este rol cumple con las siguientes características: protege el equipo de las perturbaciones externas, entrenador, facilitador, moderador y anima al equipo constantemente como líder del grupo Scrum.

[23] Jefferson Hanley (2015), Scrum: Your Quick Start Guide To Adopting Scrum For Your Organization.

Al inicio del proyecto, el Scrum Master por lo general no contribuye directamente en las metas reales del Sprint. Su principal objetivo será dar forma al equipo para conseguir que se centren en cumplir las metas comprometidas.

Idealmente el Scrum Master debe tener la confianza de todos los miembros del equipo y no debe tener ningún conflicto de interés, cuando se trata de su relación con cualquiera de los miembros del equipo.

Plantilla de Lista de Chequeo Scrum

Lista de Chequeo Scrum

Recomendado pero no siempre necesario

La mayoría de esto se necesitará, pero no siempre necesitará todo. ¡Experimente!

- | | |
|---|---|
| <input type="checkbox"/> El equipo tiene todas las habilidades necesarias para llevar los ítems del backlog a Hecho | <input type="checkbox"/> ítems del PBL se dividen en tareas dentro de un sprint |
| <input type="checkbox"/> Miembros del equipo no encerrados en roles específicos | <input type="checkbox"/> Las tareas del sprint son estimadas |
| <input type="checkbox"/> Iteraciones condenadas al fracaso son terminadas temprano | <input type="checkbox"/> Estimaciones para tareas en ejecución actualizadas diariamente |
| <input type="checkbox"/> PO tiene la visión del producto que está sincronizada con el PBL | <input type="checkbox"/> Se mide la Velocidad |
| <input type="checkbox"/> PBL y la visión del producto son altamente visibles | <input type="checkbox"/> Todos los ítems en el plan del sprint tienen un estimado |
| <input type="checkbox"/> Todos en el equipo participan en la estimación | <input type="checkbox"/> PO usa la velocidad para planeación de versiones |
| <input type="checkbox"/> PO disponible cuando el equipo está estimando | <input type="checkbox"/> La velocidad solo incluye ítems que están Hechos |
| <input type="checkbox"/> Se estima tamaño relativo (puntos de historia) en vez de tiempo | <input type="checkbox"/> El equipo tiene un gráfico de trabajo pendiente del sprint |
| <input type="checkbox"/> Todo el equipo conoce los 1-3 impedimentos principales | <input type="checkbox"/> Altamente visible |
| <input type="checkbox"/> SM tiene una estrategia para solucionar los impedimentos | <input type="checkbox"/> Actualizado diariamente |
| <input type="checkbox"/> SM se enfoca en remover los impedimentos | <input type="checkbox"/> Scrum Diario es cada día, a la misma hora y mismo lugar |
| <input type="checkbox"/> Escalado a gerencia cuando el equipo no lo puede resolver | <input type="checkbox"/> PO participa al menos unas pocas veces por semana |
| <input type="checkbox"/> El equipo tiene un Scrum Master (SM) | <input type="checkbox"/> Máximo 15 minutos |
| <input type="checkbox"/> SM se sienta con el equipo | <input type="checkbox"/> Cada miembro del equipo conoce lo que los demás están haciendo |

Escalamiento

Estas son fundamentales a cualquier esfuerzo de escalamiento Scrum.

- ☐ Tiene un Dueño de Producto en Jefe (si hay muchos PO)
- ☐ Equipos dependientes hacen Scrum de Scrums
- ☐ Equipos dependientes se integran en cada sprint

Indicadores Positivos

Principales indicadores de una buena implementación de Scrum.

- ☐ ¡Divertido! Alto nivel de energía.
- ☐ El sobretrabajo es raro y ocurre voluntariamente
- ☐ Discutir, criticar y experimentar con el proceso

PO = Dueño de Producto SM = Scrum Master PBL = Backlog de Producto DoD = Definición de Hecho
<http://www.crisp.se/scrum/checklist> | Versión 2.2 (2010-10-04)

Lo Fundamental

Si logra esto puede ignorar el resto de la lista. Su proceso está bien.

☐ Entregar software funcionando y probado cada 4 semanas o menos

☐ Entregar lo que el negocio necesita más

☐ El proceso está mejorándose continuamente

☐ Dueño del producto definido claramente (PO)

☐ PO empoderado para priorizar

☐ PO tiene el conocimiento para priorizar

☐ PO tiene contacto directo con el equipo

☐ PO tiene contacto directo con los interesados

☐ PO habla como una voz (en caso que PO sea un equipo)

☐ El equipo tiene un backlog del sprint

☐ Altamente visible

☐ Actualizado diariamente

☐ De propiedad exclusiva del equipo

☐ El Scrum Diario ocurre

☐ El equipo completo participa

☐ Problemas e impedimentos son mencionados

☐ Demo ocurre después de cada sprint

☐ Muestra software funcionando y probado

☐ Retroalimentación recibida de interesados y PO

☐ Tiene Definición de Hecho (DoD)

☐ DoD alcanzable dentro de cada iteración

☐ El equipo respeta la DoD

Scrum Esencial

Esto es central a Scrum. Sin esto probablemente no debería llamarlo Scrum.

☐ La Retrospectiva ocurre al final de cada sprint

☐ Resultados en propuestas concretas de mejoramiento

☐ Algunas propuestas llegan a implementarse

☐ Equipo completo + PO participan

☐ PO tiene backlog de producto (PBL)

☐ Ítems son priorizados por su valor al negocio

☐ Ítems son estimados

☐ Estimaciones hechas por el equipo

☐ Ítems en PBL tan pequeñas para hacerse en un sprint

☐ PO entiende el propósito de todos los ítems del backlog

☐ Tienen reuniones de planeación del sprint

☐ PO participa

☐ PO lleva el PBL actualizado

☐ El equipo completo participa

☐ Resultados en un plan del sprint

☐ El equipo completo cree que el plan es alcanzable

☐ PO satisfecho con prioridades

☐ Iteraciones tiempo-fijo

☐ Longitud de iteración de 4 semanas o menos

☐ Siempre terminan a tiempo

☐ Equipo no interrumpido o controlado por externos

☐ Equipo usualmente entrega lo que se comprometió a hacer

☐ Miembros del equipo se sientan juntos

☐ Máximo 9 personas por equipo

Product Owner (Dueño de producto): Esta es una combinación de la función desempeñada por el tradicional gerente del producto y gerente de proyecto. “El Product Owner actúa como intermediario entre el Equipo Scrum y el cliente, es responsable de asegurarse de que el equipo está trabajando en las entregas correctas en el momento adecuado” [24].

Además de trabajar en estrecha colaboración con el Equipo Scrum en otros roles acordados, el Product Owner principalmente realiza lo siguiente:

Administra el Scrum Product Backlog.

Decide qué entregables deben ser producidos y cuando.

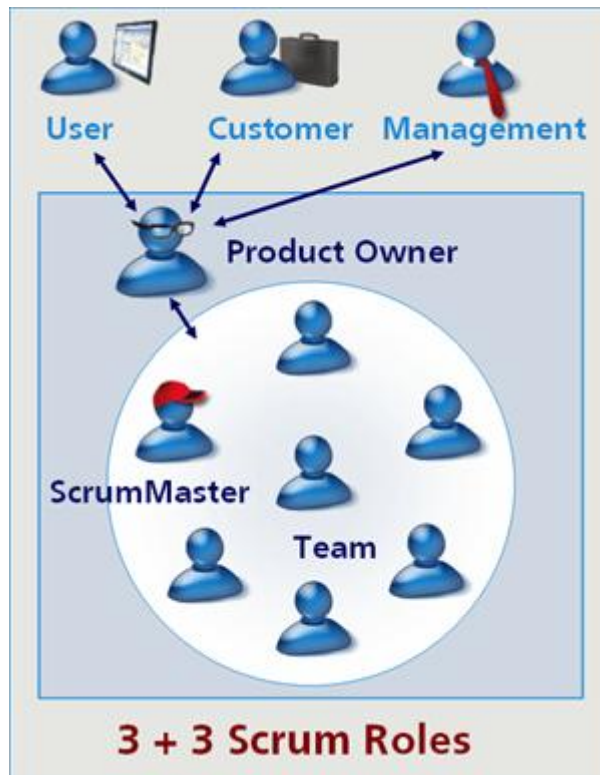
Controla el calendario de lanzamiento de los productos entregables terminados.

Busca cualquier aclaración de fuentes externas para el Equipo Scrum sobre el Product Backlog.

La comunicación entre el equipo Scrum y los clientes es un papel primordial del Product Owner. Por lo tanto se debe realizar por alguien con habilidades de comunicación escrita y verbal. Además la realización de esta función requiere de alguien con buenas habilidades de gestión de negociación. [25]

[24] [25] Jefferson Hanley (2015), Scrum: Your Quick Start Guide To Adopting Scrum For Your Organization.

Figura 6. Product Owner



Fuente: <http://www.barisdere.com/2010/12/cevik-surecler-agile-scrum/>

Las Estimaciones del Backlog

Antes de la primera reunión de planificación el Equipo tiene que conocer cuál va a ser su velocidad inicial y su factor de dedicación.

Para poder realizar las estimaciones, primeramente hay que decidir qué historias incluir en el Sprint Backlog.

La forma en la que se podrá decidir qué historias poner o no se puede realizar mediante 2 técnicas: De forma aproximada y realizando cálculos de velocidad.

De forma aproximada:

En la estimación aproximada, el equipo debate el número de historias a introducir hasta llegar a un acuerdo. Suele funcionar de forma correcta si los Sprints son cortos.

Realizando los cálculos de velocidad:

Esta técnica se realiza en dos pasos:

- Seleccionar la velocidad estimada.

- Calcular el número de historias que se pueden añadir sin pasar la velocidad estimada.

La manera más adecuada de estimar la velocidad, es revisar el historial del equipo, de esta manera, basándonos en Sprints anteriores se podrá deducir que será muy similar.

Para poder realizar esta técnica, es necesario que los equipos tengan un historial, que vayan hacer los próximos Sprints de la misma manera, mismo tamaño de equipo y las mismas condiciones de trabajo. La técnica se conoce como “el tiempo que hizo ayer”.

Las características del Backlog del producto serán las siguientes:

La estimación será entre 4 y 16 horas, si son mayores a este valor, se descompondrán.

Las tareas del Sprint deben de ser como consecuencia de la necesidad de un requerimiento del Backlog del producto.

El progreso de las tareas en las que se mide la velocidad y el progreso del Sprint con respecto a las horas, se realizará mediante gráficos Burn down Chart.

El soporte en el que se presentan el Sprint Backlog puede ser:

- Hoja de cálculo.

- Pizarra.

Herramientas colaborativas de red.

La herramienta como soporte quizás más utilizado es el Scrum Taskboard en la que en una pizarra se crea una tabla con la siguiente información:

1ª fila: Tareas que no se han planificado pero que son necesarias de hacer por su urgencia (errores, cambios importantes decididos por el cliente, etc).

2ª fila: Mejora continua. Se ponen las tareas de retrospectivas anteriores que se quieren analizar en ese Sprint.

Columna tareas no empezadas: Se pondrán todas las tareas que se han especificado en la reunión del Sprint planning.

En curso: Tareas que se están realizando y que deberían ser mínimas y resueltas de arriba abajo.

Hecho: Tarea realizada completamente.

Impedimentos: Lista de obstáculos que impedirán continuar de forma adecuada el proyecto. Hay que indicar quién será el responsable de solucionarlos.

Retrospectiva: Sirve para anotar qué partes están bien durante la iteración y cuáles mal. Se reflejan con un "+" y un "-". [26]

[26] Manuel Trigas Gallego, Gestión de proyectos informáticos, metodología Scrum. pag. 32

El equipo Scrum: Además de lo dicho anteriormente acerca de equipos Scrum en general, estos equipos tienen otras características que los diferencian de los equipos que utilizan otras metodologías de gestión de proyectos.

Segun Trigas [27] la esencia del equipo Scrum tienen las siguientes características:

Es relativamente pequeño por lo general entre 5 a 8 individuos.

No tiene sub equipos, que funcionan como una entidad en sí misma.

Trabaja con alto grado de autonomía como una unidad auto organizada.

Son autos gobernados tomando su dirección internamente dentro del equipo y las reglas y normas que han acordado adoptar.

Trabaja mejor con personal con disponibilidad de tiempo completo.

[27] Manuel Trigas Gallego, Gestión de proyectos informáticos, metodología Scrum.

El equipo es responsable de las siguientes tareas:

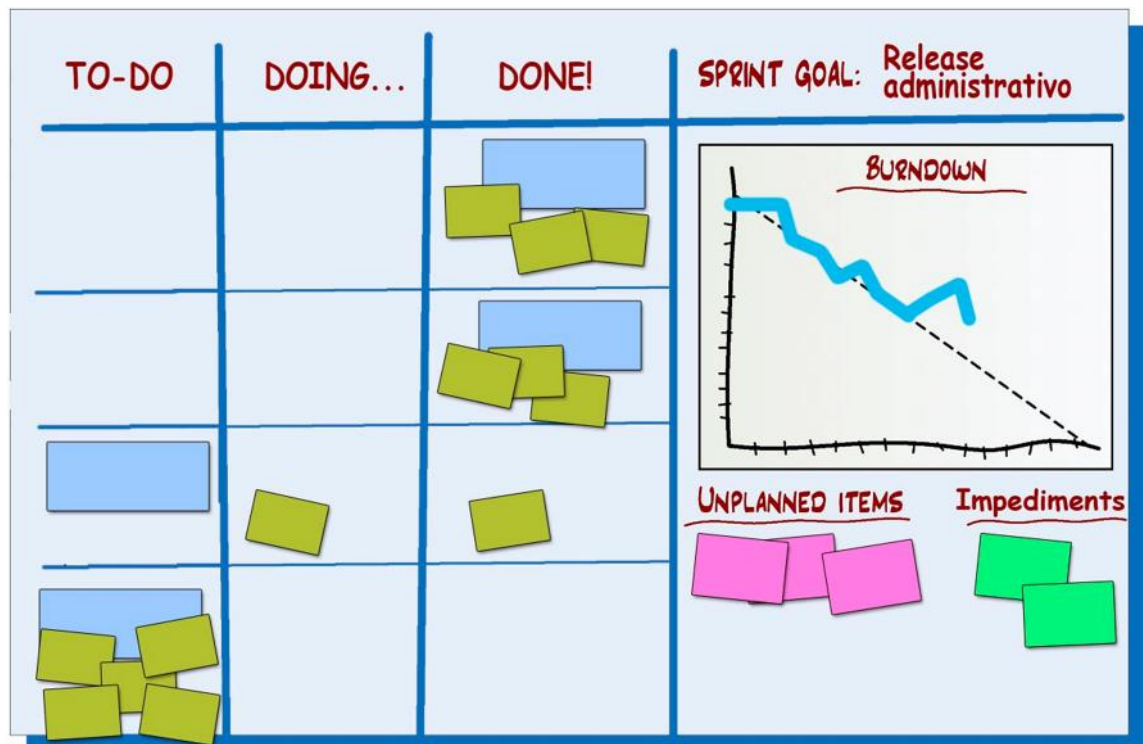
La producción y mantenimiento de ciertos artefactos Scrum principalmente el Product Backlog.

La organización, el funcionamiento y la participación en la Reunión diaria del Sprint.

Trabajar colectivamente para producir un entregable funcional al final de cada Sprint.

Asegurarse que las actualizaciones de estado en el Taskboard (tablero) del equipo, específicamente el **Sprint Burn down Chart** (Gráfica de avance del Sprint), es regular y precisa.

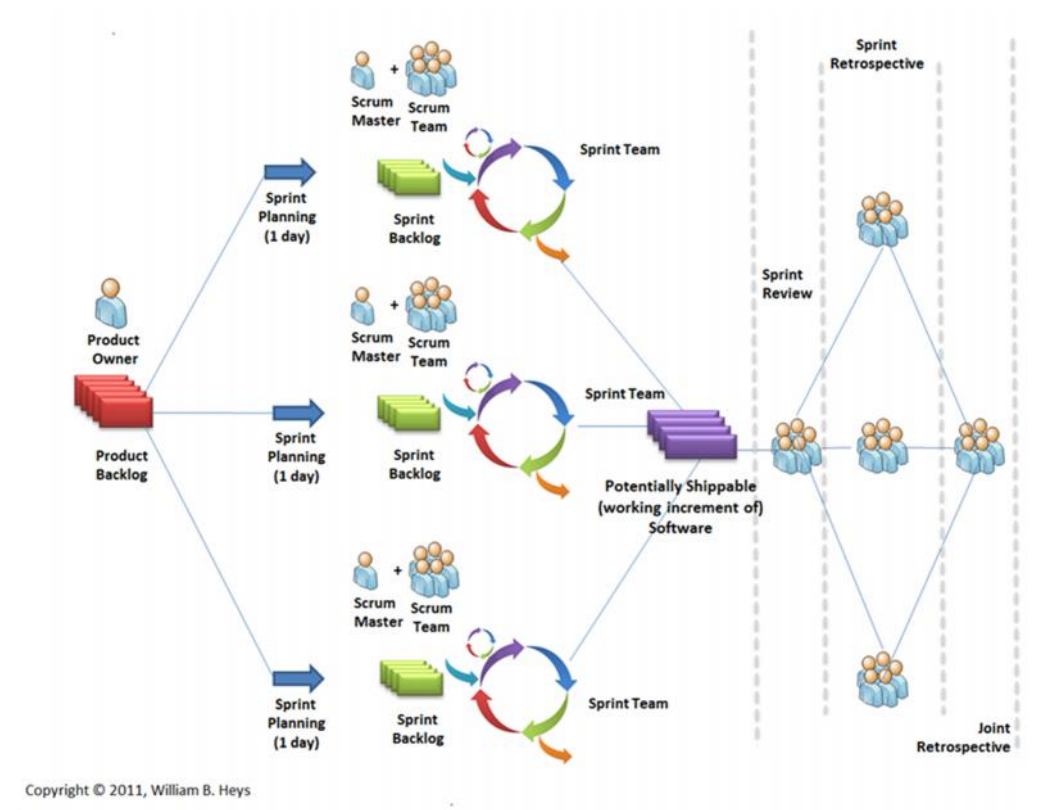
Figura 7. Taskboard



Fuente: pitalab.com

6.4.3.3 Eventos Scrum: Los proyectos Scrum se llevan a cabo a través de una serie de eventos que están orientados a la producción de todos los entregables del proyecto. Además hay otros eventos de Scrum que sirven para gestionar los plazos y la calidad de las prestaciones así como realizar evaluaciones del Scrum en general, los procesos que se siguieron y adoptar medidas para simplificar cuando sea necesario.

Figura 8. Eventos Scrum

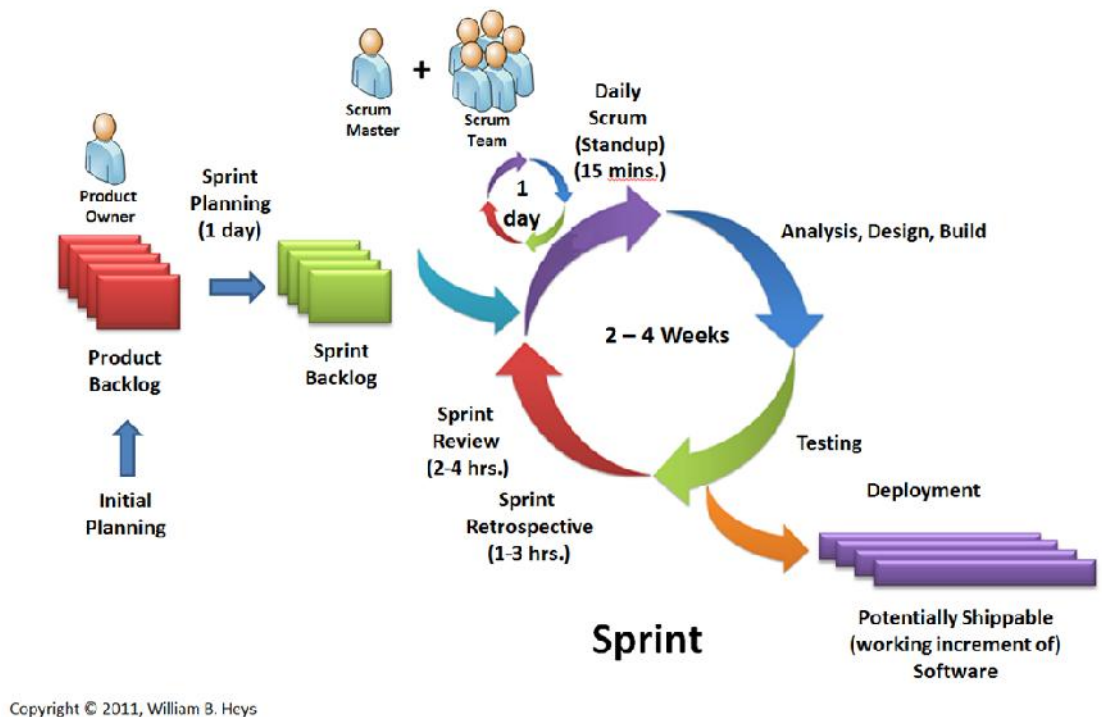


Fuente: <http://blogs.msdn.com/b/billheys/archive/2011/01/18/branching-for-scrum.aspx>

Sprint: Según dice Hanley [28] en la estructura de Scrum, todas las actividades de los proyectos diseñados para ofrecer elementos del mismo como Product Backlog se realizan a través de un evento conocido como el Sprint o iteración. Los Sprints suelen limitarse a duraciones entre 1 y 4 semanas.

El objetivo del sprint es crear las condiciones ideales para que el equipo de trabajo Scrum logre terminar todos los objetivos que se tenga en el Sprint Backlog.

Figura 9. Sprint

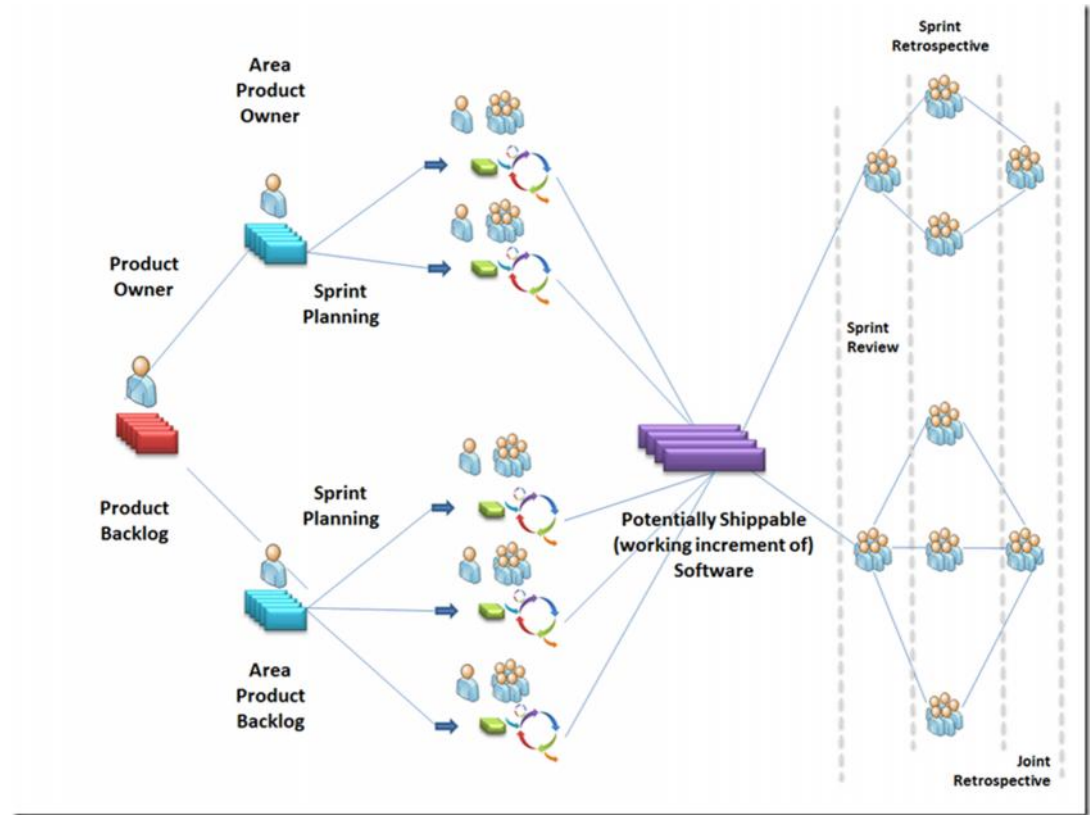


Fuente: <http://blogs.msdn.com/b/billheys/archive/2011/01/18/branching-for-scrum.aspx>

[28] Jefferson Hanley (2015), Scrum: Your Quick Start Guide To Adopting Scrum For Your Organization.

Sprint Planning:

Figura 10: Sprint planning



Fuente: <http://blogs.msdn.com/b/billheys/archive/2011/01/18/branching-for-scrum.aspx>

Segun Hanley [29] Sprint planning es una reunión, en la que participarán el Product Owner, el Scrum Master y el equipo Scrum, con la intención de seleccionar de la lista Backlog del producto las funcionalidades sobre las que se va a trabajar y que darán valor al producto.

[29] Jefferson Hanley (2015), Scrum: Your Quick Start Guide To Adopting Scrum For Your Organization.

Antes de comenzar la reunión el Product Owner tendrá que preparar el Backlog. La reunión se realizará durante ocho horas dividido en 2 partes de 4 horas.

Primera parte de la reunión:

El equipo selecciona los ítems para transformarlos en entregables.

El equipo hace sugerencias, pero es el Product Owner el que decidirá si formarán parte del Sprint.

El equipo seleccionará el elemento a implementar de los seleccionados por el Product Owner para ese Sprint.

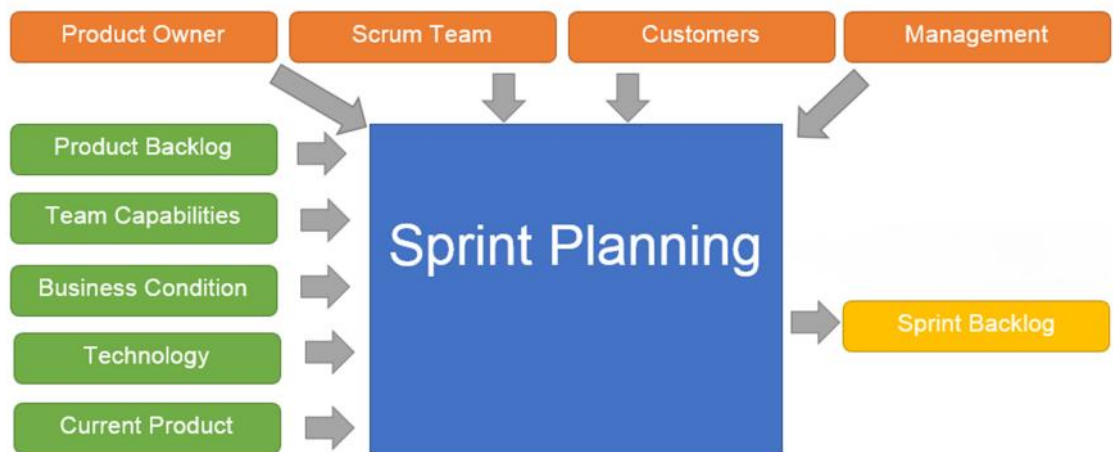
Segunda parte de la reunión:

El equipo hará las preguntas necesarias que tengan sobre el Product Backlog al Product Owner.

El equipo se encargará de encontrar la solución adecuada para transformar la parte seleccionada en una funcionalidad entregable.

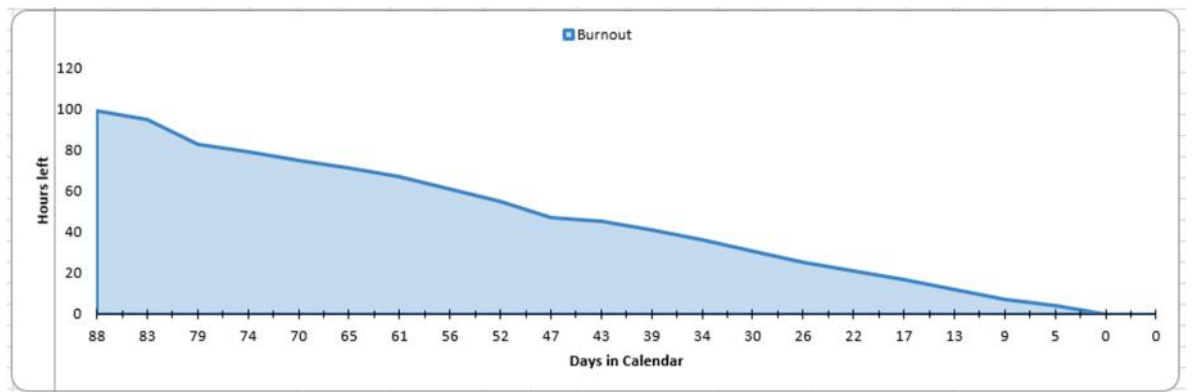
El resultado de la segunda parte de la reunión es una lista denominada Sprint Backlog con las tareas, estimaciones y las asignaciones de trabajo al equipo para poder empezar a desarrollar la funcionalidad.

Figura 11. Entradas / Salidas de un Sprint Planning



Plantilla de Planificación del Sprint

Task	Time (estimated)	Time (spent)	Time (left)	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
Planning	6	6	0	4	1	1																			
Team documentation	4	4	0		3	1																			
Wireframing	4	11	0		8	2	1																		
UI	3	3	0				3																		
Design	8	8	0					4	4																
HTML/CSS	6	8	0						6	2															
Theme development	12	12	0							4	8														
Server setup	4	4	0									2	2												
Testing 1	3	3	0										2	1											
Bug fixes	4	5	0											4	1										
Optimization	4	4	0													4									
Content	5	6	0														3	3							
SEO	10	10	0														2	1	4	3					
User documentation	8	8	0														1			2	3	1	1		
Go live	2	2	0																						
Testing 2	5	5	0																						
		0	0																						
		0	0																						
TOTAL	88	99	0	4	12	4	4	4	4	6	8	2	4	5	5	5	4	4	5	4	4	5	3	4	0
Daily burnout	0			5	4	5	4	5	4	5	4	5	4	4	5	4	4	4	5	4	4	4	4	5	0
Total time left (from estimate)			88	83	79	74	70	65	61	56	52	47	43	39	34	30	26	22	17	13	9	5	0	0	0
Total time left (from spent)			99	95	83	79	75	71	67	61	55	47	45	41	36	31	25	21	17	12	7	4	0	0	0



Daily Scrum (Reunión diaria): Según dice Hanley [30] ahora que el Equipo Scrum ha recibido las tareas que necesita ser entregadas al finalizar el Sprint, el equipo se reúne diariamente para gestionar su carga de trabajo y garantizar que el progreso está siendo realizado. El Scrum Master facilita los Daily Scrum y la participación de todos los miembros del equipo es obligatoria. Estas reuniones se realizan de pie y no debe exceder los 15 minutos de duración. Si hay elementos significativos durante esta sesión no deben ser ignorados debido a la falta de tiempo.

[30] Jefferson Hanley (2015), Scrum: Your Quick Start Guide To Adopting Scrum For Your Organization.

Las reuniones Off Line podrían programarse para hacer frente a este tipo de aspectos por separado.

Durante el Daily Scrum, el Scrum Master revisará lo siguiente con cada miembro del equipo:

Lo que se llevó a cabo desde la reunión Daily Scrum anterior

¿Qué le impidió cumplir con los objetivos?

¿Qué es lo que van a llevar a cabo antes de la próxima sesión?

El Sprint Backlog se actualiza como resultado de estas sesiones diarias para reflejar el estado actual del Sprint.

Estas reuniones son beneficiosas para todos, puesto que se comprueba:

Si el equipo ha cumplido las expectativas.

Y si el equipo ha entendido al cliente.

Sprint Review (Revisión del Sprint): Según Hanley [31] ya que cada Sprint tiene la intención de culminar con la finalización de ciertas entregas, esto implica que la revisión del Sprint se llevará a cabo una vez que esas entregas se hayan completado.

[31] Jefferson Hanley (2015), Scrum: Your Quick Start Guide To Adopting Scrum For Your Organization.

Durante estas sesiones que se celebran al final de un Sprint, el Equipo Scrum demuestra todos los elementos del Sprint Backlog terminados. El Product Owner es responsable de aceptar o rechazar cualquier artículo del Sprint Backlog. En caso de ser rechazado un elemento, el Product Owner lo eliminará del Sprint Backlog. Tales aspectos son revisados y re-priorizados para su posible inclusión en Sprints posteriores.

La revisión del Sprint debe mantenerse lo más informal posible con menos énfasis en la ceremonia y más en la sustancia, tales como las decisiones que indican el estado de elementos del Sprint Backlog: aceptada, rechazado o necesita mejoras.

Sprint Retrospective (Retrospectiva del Sprint): Según lo dice Hanley [32] como su nombre lo indica se trata de una mirada retrospectiva al Sprint recientemente terminado que tendrá una duración de máxima de 3 horas donde asistirán el ScrumMaster, el Equipo Scrum y el Product Owner. La reunión Sprint Retrospective debe llevarse a cabo inmediatamente después de la finalización de un Sprint y debe centrarse en las siguientes tres cosas:

Qué es lo que el Equipo Scrum hizo bien durante el Sprint?

Qué es lo que no salió según lo planeado durante el Sprint?

¿Qué mejoras se necesitan hacer para que las cosas puedan ir aún mejor en el próximo Sprint?

[32] Jefferson Hanley (2015), Scrum: Your Quick Start Guide To Adopting Scrum For Your Organization.

El Scrum Master facilita estas sesiones y es obligatorio para todo el Equipo Scrum asistir y proporcionar retroalimentación. El Scrum Master documenta estas sesiones y actualiza las lecciones aprendidas y mejora la documentación para futuras consultas, también toma nota de las respuestas del Equipo en un formulario.

Si hay puntos que merezca atención se añadirán para el siguiente Sprint, considerándolo como un Backlog no funcional, pero de alta prioridad.

6.4.3.4 Artefactos de Scrum: Como bien lo dice Hanley [33] todo buen proceso o metodología, Scrum propone ciertas herramientas llamadas Artefactos de Scrum que ayudan a los profesionales a documentar el proyecto en general. El Equipo Scrum utiliza estos artefactos como ayudas visuales para gestionar y realizar un seguimiento de los progresos que hizo en el proyecto así como en los Sprints.

Product Backlog: Es el inventario en el que se almacenan todas las funcionalidades o requisitos en forma de lista priorizada. Estos requisitos serán los que tendrá el producto o los que irá adquiriendo en sucesivas iteraciones.

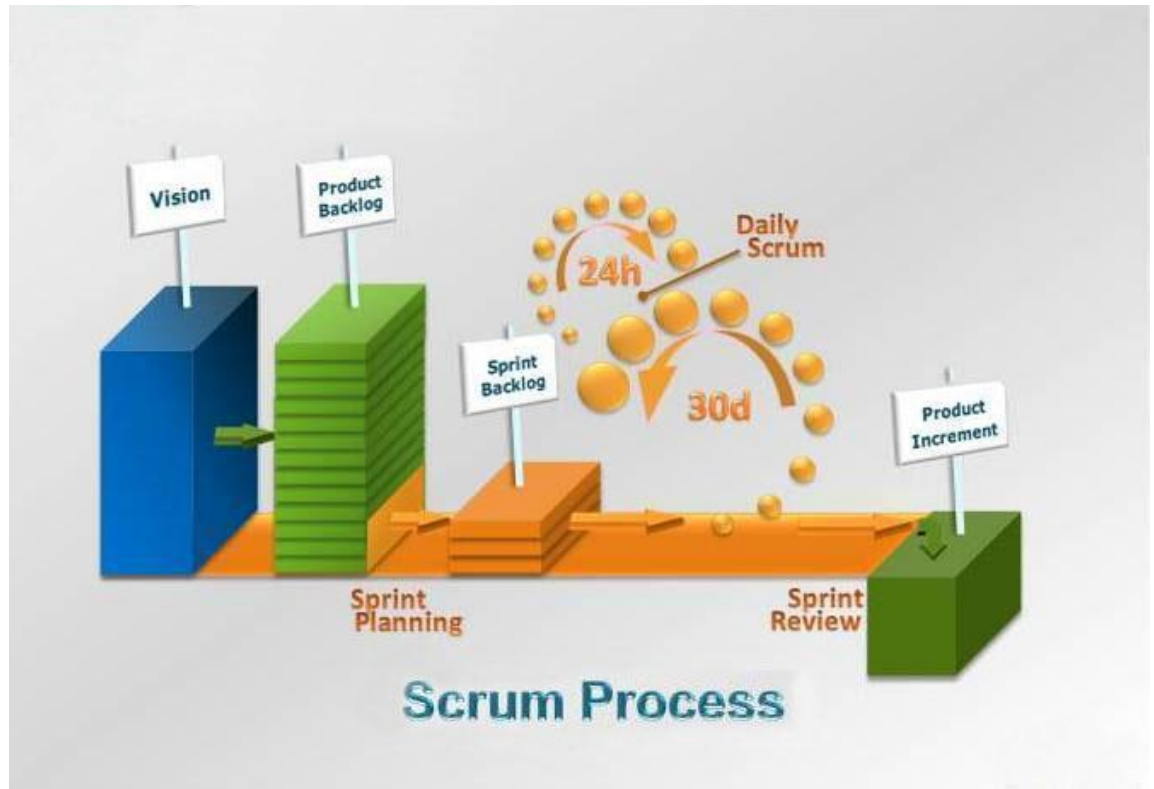
[33] Jefferson Hanley (2015), Scrum: Your Quick Start Guide To Adopting Scrum For Your Organization.

Todos los equipos de proyectos independientemente de la metodología que siguen para gestionar el proyecto, necesitan algún tipo de lista de características para trabajar. En su forma más simple, el Scrum Product Backlog se puede considerar como el universo entero de características que debe ser entregado a fin de considerar el proyecto terminado.

Es importante señalar que el Product backlog va mucho más allá de una simple lista por hacer. Esta herramienta clasifica y prioriza todas las características del producto que son de alcance en el proyecto. Más importante aún, proporciona un enlace a los casos de uso específicos, así como al Sprint particular en el que la función será o ha sido entregado.

El Product Backlog sirve como un artefacto importante durante el Sprint Planning, ya que se utiliza para identificar qué características deben ser incluidas en el próximo Sprint. El Product Owner posee y mantiene Product backlog y lo utiliza para seguir el progreso de los entregables de cada Sprint.

Figura 12. Origen del Product Backlog

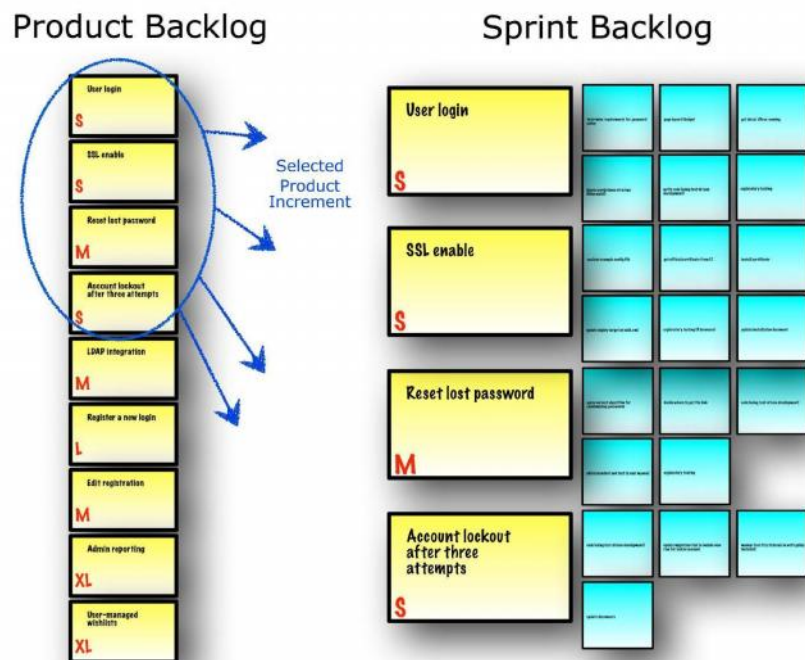


Fuente: <http://guntherverheyen.com/category/agile/my-fragility-agile/page/2/>

Sprint Backlog: Según Hanley [34] los elementos del Product Backlog una vez especificados son seleccionados para su inclusión en un Sprint, el desafío consiste en controlar y realizar un seguimiento de todas las tareas necesarias para entregarlos durante el Sprint. El Sprint Backlog es otra poderosa herramienta que ayuda a los equipos Scrum a definir lo más importante que se necesita para lograr sus objetivos que serán realizados en el siguiente Sprint.

[34] Jefferson Hanley (2015), Scrum: Your Quick Start Guide To Adopting Scrum For Your Organization.

Figura 13. Sprint backlog



Fuente: <http://scrumreferencecard.com/scrum-reference-card/>

Segun Hanley [35] Sprint Backlog ofrece una imagen visual de:

El PBI (Product Backlog) que están programadas para la entrega durante el Sprint actual.

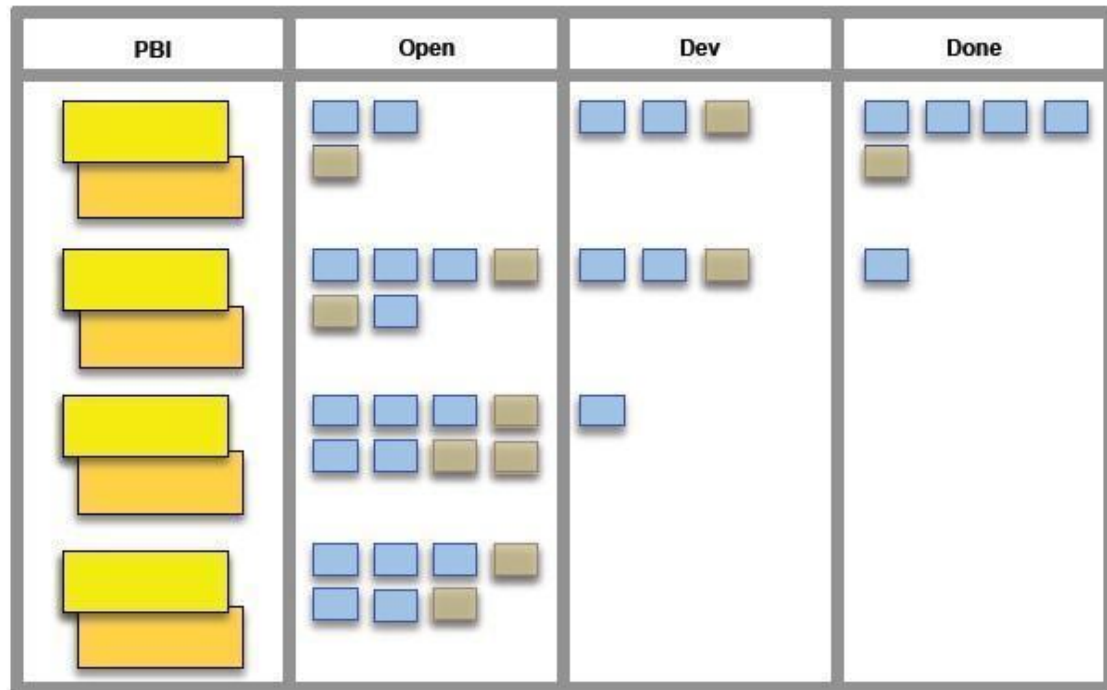
La lista de tareas abiertas que deben completarse para que cada uno de los elementos elegidos del PBI puedan ser considerados hechos.

Un subconjunto de esas tareas abiertas que se han asignado a un desarrollador y están siendo activamente trabajadas sobre él.

[35] Jefferson Hanley (2015), Scrum: Your Quick Start Guide To Adopting Scrum For Your Organization.

Una lista de tareas de desarrollo que se han completado cuando un miembro del equipo se le asigna una tarea, esta se elimina de la cola abierta y entra en la cola en desarrollo, con su nombre sobre ella. Cuando se termina una tarea, se retira de en desarrollo y se añade a la cola realizado.

Figura 14. Desarrollo del Sprint backlog



Fuente:

<http://www.methodsandtools.com/archive/atddreadysprintbacklog.php>

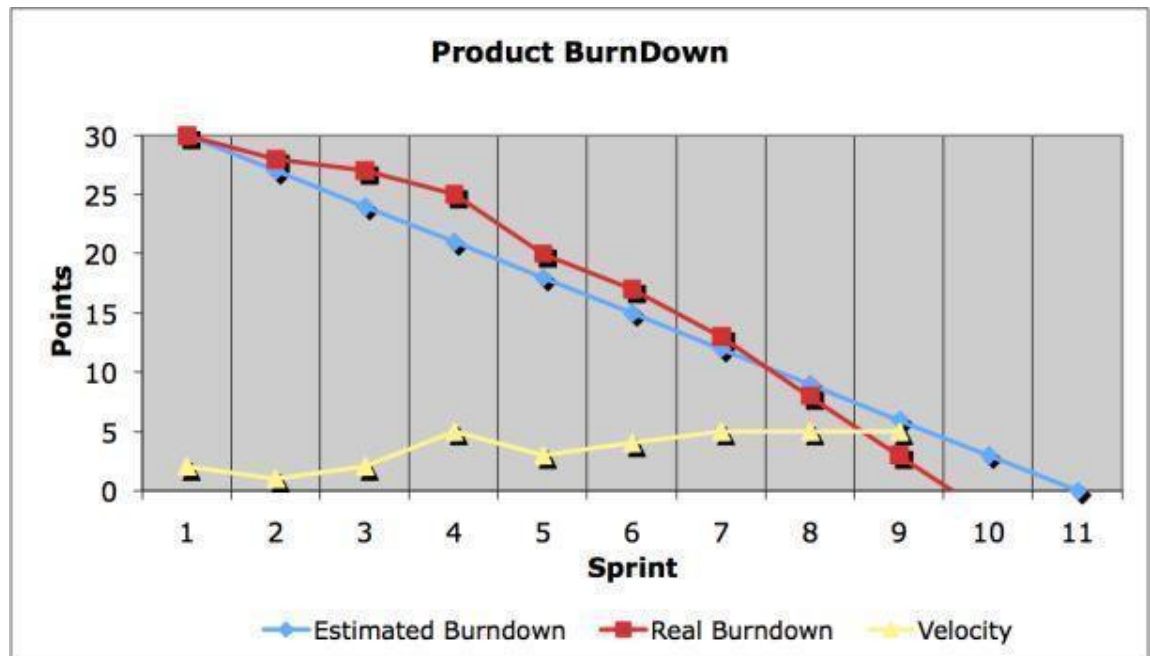
Equipos de Scrum poseen y administran el Sprint Backlog y se actualiza todos los días (a menudo varias veces al día). Se revisó durante el Daily Scrum, seguida de actualizaciones a la vista de todos. Por lo general es de gran ayuda para que el equipo también estime el tiempo para completar y el tiempo que queda hasta el final en horas hombre para que todos los miembros del equipo y las

partes interesadas tengan una clara indicación de los trabajos en curso y el trabajo que falta. [36]

Plantilla de Sprint Backlog

Ciclo Sprint 1												
Fecha Inicio		02/02/2009		Grupo		Aplicaciones Móviles						
Fecha Fin		16/02/2009										

Figura 15. Burn down chart



Fuente: http://www.scrum-institute.org/Burndown_Chart.php

Esta herramienta ayuda a los gerentes de proyectos y otras partes interesadas ver exactamente lo rápido que el Equipo Scrum está trabajando a través del product Backlog.

Características que cumple el Burn Down chart:

Es una ayuda visual que muestra los progresos realizados completando varios objetivos para entregar el producto final.

En él se compara cómo el equipo está haciendo el trabajo en términos de esfuerzo en la finalización de los entregables.

Proyecta que la finalización se verá de una forma, si se mantiene el ritmo actual de progreso.

Traza la velocidad del avance del Equipo Scrum.

Los gerentes de proyectos pueden usar varios formatos de Burn Down chart centrándose en varias métricas, tales como el esfuerzo total

en comparación con el trabajo realizado; Esfuerzo total frente Trabajo restante; Esfuerzo frente a la velocidad. Estas presentaciones ayudarán al Scrum Master y el Product Owner decidir cómo manejar el progreso, o cambiar la prioridad del Product backlog de una manera más realista.

Los Burn Down charts deben ser revisadas durante cada Daily Scrum y los miembros del equipo deben comentar por qué las cosas no están progresando según lo planeado (si eso está sucediendo).

6.4.3.5 Reglas de Scrum: Según lo dice Hanley [38] las reglas de Scrum son el pegamento que une al equipo Scrum a los roles, eventos y artefactos dentro de la estructura general de Scrum. Mientras que algunas metodologías tienen reglas extremadamente rígidas, lo que diferencia a Scrum es que son más incluyentes y flexibles en su naturaleza.

Si bien algunas de estas reglas son directrices para la adopción de las mejores prácticas de Scrum, otros adoptan la forma de normas generales que deben promoverse en un ajuste del proyecto. En conjunto, estas reglas y normas a continuación regirán la interacción entre los miembros del equipo.

He aquí algunas reglas clave de Scrum que los practicantes deben tener en cuenta:

El tiempo de los Sprint deben ser de la misma duración.

Los Sprint no deben exceder 4 semanas de duración.

Los equipos no deben tomar descansos entre Sprint, deben seguir un ciclo continuo.

[38] Jefferson Hanley (2015), Scrum: Your Quick Start Guide To Adopting Scrum For Your Organization.

Todo el equipo debe entender que el objetivo final de un Sprint es crear un producto potencialmente entregable.

Cada Sprint debe comenzar con un Sprint planning.

El Sprint planning deben ser de 2 a 3 horas.

Cada día del Sprint debe comenzar con un Daily Scrum

Los Daily Scrum deben ser de 15 minutos como máximo.

Todos los miembros del equipo deben asistir al Daily Scrum.

La fecha, hora y lugar de cada reunión deberá ser publicada con anterioridad.

Cada Sprint debe concluir con un Sprint Review.

Cada Sprint Review debe ser seguido por un Sprint Retrospective y todos los miembros del equipo deben asistir.

Al priorizar los elementos del Product backlog, no hay dos elementos que tengan la misma prioridad.

Si hay algunos defectos conocidos en el Sprint anterior, deben priorizarse para su resolución al más alto nivel en el próximo Sprint.

Las reuniones del equipo deben ser abiertas y equitativas con todos los miembros y las partes interesadas que permitan expresar sus puntos de vista.

En los equipos Scrum es recomendable que use pantallas visuales o taskboard bien posicionadas para resaltar el progreso.

Cuando un miembro del equipo se enfrenta con dificultad a un reto, deberá buscar activamente la ayuda de otras personas que son capaces de apoyarlo.

Los miembros del equipo Scrum que completen sus tareas antes de tiempo debe ser voluntario para tareas adicionales del Sprint backlog.

Si bien las reglas ayudan a poner orden y disciplina dentro de los equipos de proyecto, ninguna cantidad de ellas puede realmente prevenir algunos posibles tropiezos.

Los principios clave detrás de las reglas Scrum, son los siguientes:

Todo el mundo debe ser tenido en cuenta antes de proponer una regla.

Todos los miembros del equipo y las partes interesadas deben estar de acuerdo en cumplir las normas.

Las reglas deben ser bien publicitadas.

Debe haber consecuencias por no seguir una regla.

En última instancia, estas reglas y normas son directrices de alto nivel a la hora de gestionar proyectos.

6.5 VIVIR CON SCRUM, LOS CASOS DE USO

Segun Hanley [39] uno de los retos más difíciles para los Scrum Masters es tener que hacer frente a la hora de definir exactamente lo que el usuario desea para las entregas. El reto es aún más significativo en Scrum porque cada Sprint se espera terminar con la entrega de los módulos con los requisitos al pie de la letra. Por lo tanto si no se entiende desde el primer Sprint lo que en realidad hay que hacer se terminaría produciendo un producto inservible.

Por suerte, el Sprint tiene una herramienta muy poderosa para ayudar a los profesionales a definir estos requisitos en alto grado de detalle. La herramienta se llama la historia de usuario.

[39] Jefferson Hanley (2015), Scrum: Your Quick Start Guide To Adopting Scrum For Your Organization.

6.5.1 ¿Qué es una historia de usuario? En pocas palabras, Según Hanley [40] una historia de usuario no es más que un caso de uso que ilustra la necesidad y la utilidad de una característica específica o funcionalidad del producto. Como su nombre lo indica, las historias de los usuarios son impulsados por el usuario final, pero su creación se debe facilitar por el Product Owner.

El Product owner en conjunto con los usuarios elaboran un conjunto exhaustivo de historias de los usuarios que definen el producto en su totalidad. Una vez que todas las historias de los usuarios se compilan pueden ser cedidos a otras funciones del equipo Scrum para la actividad de transformación.

El sello distintivo de proyectos Scrum es que la metodología permite a los equipos de Scrum adaptarse rápidamente a las cambiantes historias de los usuarios y dar cabida a los cambios en Sprints siguientes en lugar de trabajar en un nuevo proyecto. La gestión exitosa de historias de los usuarios es una parte muy importante de cómo se garantizará el éxito del proyecto Scrum.

[40] Jefferson Hanley (2015), Scrum: Your Quick Start Guide To Adopting Scrum For Your Organization.

6.6 ¿PORQUE SCRUM?

Segun lo dice Hanley [41] el uso de cualquier estructura de gestión de proyectos que son probadas para ejecutar proyectos puede generar beneficios significativos en toda la organización. Los beneficios pueden incluir:

- La alta dirección
- Miembros de los empleados
- Clientes
- Jefes de Proyecto
- Desarrolladores
- Gerentes de Producto ... y más

Sin embargo, la utilización de Scrum proporciona mejores beneficios que las metodologías de gestión de proyectos convencionales podrían no ofrecer:

Un aumento en el retorno de inversión de las organizaciones mediante la producción de un conjunto estable y continuo de los beneficios de valor agregado en lugar de la entrega de un producto al final de un proyecto.

Dado que todos los actores están continuamente y constantemente ocupados con el proyecto a través de Sprints exitosos, el resultado es más confiable que una que se relaciona con las partes interesadas en la reunión de toma de requisitos al inicio y al final con las pruebas de aceptación.

El riesgo general del proyecto y la incertidumbre se reduce drásticamente como resultado de descomponer los entregables en fracciones más pequeñas y casi autónomas.

[41] Jefferson Hanley (2015), Scrum: Your Quick Start Guide To Adopting Scrum For Your Organization.

Mediante la producción del entregable al final de los Sprint, el tiempo de mercado de la organización aumenta porque los usuarios no tienen que esperar hasta que el producto esté totalmente hecho antes de la ejecución.

Un equipo Scrum bien organizado es un grupo ágil y eficiente que ofrece un trabajo productivo, creativo e innovador en la organización.

El resultado final de la utilización de Scrum es que todo el mundo en la organización se da cuenta de los beneficios que conlleva usarlo.

6.6.1 Ajuste de expectativas: Aunque Scrum puede ser una herramienta poderosa para acelerar los proyectos de la organización a través de los promotores Scrum, no debe esperar que sea un éxito de la noche a la mañana. Por ejemplo, en las organizaciones que tienen una arraigada cultura de jerarquía y una desconfianza fuerte a las cosas nuevas, adoptar Scrum podría ser un difícil reto de alcanzar. Por lo tanto profesionales de Scrum, siempre aconsejan a los potenciales adoptantes Scrum manejar cuidadosamente las expectativas de la organización antes de proponer el uso de Scrum.

6.7 SCRUM EN ACCIÓN

Ahora que se tiene una buena comprensión de la estructura Scrum y comprendemos la importancia de los equipos Scrum, es hora de considerar los procesos que se necesitan para poner Scrum en acción.

6.7.1 La construcción del equipo: Según dice Hanley [42] en un proyecto exitoso, Scrum aportará gran parte a él. Y el éxito del equipo en sí no está en lo que cada miembro individualmente es capaz de hacer, sino en lo que el equipo es capaz de entregar colectivamente. Y ahí está el reto para los profesionales de Scrum. Uno puede tener los mejores y más brillantes individuos asignados a un proyecto, pero si no lo hacen o no son capaces de trabajar juntos por un objetivo común, su brillantez significará nada para el proyecto.

Y eso dará lugar a un nuevo desafío de equipo: Teniendo en cuenta que los equipos Scrum están desprovistos de cualquier papel formal de jefe de proyecto, ¿Cómo lograr conseguir realmente un equipo de individuos, si algunos de ellos nunca han trabajado juntos antes, para entregar con éxito todos los elementos del Product Backlog? Y la respuesta es: Mediante el uso de un poco de la psicología humana y un poco de astucia por parte del Scrum Master.

[42] Jefferson Hanley (2015), Scrum: Your Quick Start Guide To Adopting Scrum For Your Organization.

6.7.1.1 Mentalidad individual: En el mundo real, los individuos que son miembros del equipo del proyecto Scrum es poco probable que respondan a algunos de los estímulos que podría motivarlos a trabajar. Ideales como seguridad en el empleo y promociones, no son algo que pueda conducir a un miembro del equipo Scrum a dar lo mejor en el equipo. Para construir equipos Scrum eficaces tenemos que mirar en la mentalidad y punto de vista de los miembros potenciales del equipo.

En el año 1954 Abraham Maslow, el padre de la psicología humanista, se le ocurrió una pirámide que representa una jerarquía de las necesidades humanas. Según Maslow, las personas tienen 5 niveles con distintas necesidades que deben ser satisfechas. Los hallazgos de Maslow son importantes hoy en día, porque en algún lugar a lo largo de esa pirámide se encuentra lo que motiva a los individuos en un equipo Scrum. Y si se quiere construir equipos eficaces, entonces se tiene que entender la mentalidad de los posibles miembros del equipo Scrum.

Figura 16. La jerarquía de las necesidades, según maslow



Fuente: <http://circulocuadrado.es/blog/la-motivacion-humana-2a-parte/>

Citado por Hanley [43] Maslow teorizó que las necesidades fisiológicas tales como alimentos, agua y otras necesidades corporales forman la base de la pirámide de las necesidades humanas.

Luego vienen las necesidades de seguridad que en el contexto de un miembro del equipo del proyecto podría equiparar la seguridad laboral, seguridad de los ingresos, ser propietario de una casa y prosperar en un empleo estable.

Entonces después, vino la necesidad de amor y pertenencia que hablan sobre la necesidad de un individuo a tener relaciones sociales.

Los dos últimos tipos de necesidades la autoestima y autorrealización son lo que debe interesar a los profesionales de Scrum, porque eso es lo que les permite a los miembros del equipo Scrum a crecer profesionalmente. Estas son las necesidades humanas que determinan sentimientos como: La realización personal, la auto motivación, confianza, respeto por el otro, creatividad, mentalidad abierta, dedicación y compromiso

Si usted quiere poner Scrum en acción con el mayor vigor, entonces son estas necesidades individuales las que debe potenciar para lograr que los miembros del equipo Scrum puedan contribuir activamente en los objetivos de los proyectos.

[43] Jefferson Hanley (2015), Scrum: Your Quick Start Guide To Adopting Scrum For Your Organization.

6.7.1.2 Creación y gestión del equipo: Aquí están algunas sugerencias prácticas para la construcción del equipo Scrum:

Los equipos de Scrum deben sacar fuerzas de desempeño a nivel grupal. Eso significa que incluso cada uno es un gran contribuyente, si un miembro del equipo no es capaz de funcionar en un ambiente de grupo, su inclusión en el equipo Scrum podría ser potencialmente perjudicial.

Al seleccionar los miembros del equipo, se debe asegurar de que pueden ser contribuyentes importantes dentro del equipo, las personas que no tienen ningún valor agregado para ofrecer sólo serán perjudiciales para el rendimiento del equipo.

Los equipos demasiado pequeños pueden conducir a los individuos a llevar mucha carga y por lo tanto los agota rápidamente. Y un tamaño demasiado grande del equipo puede dar lugar a ineficiencias. El tamaño óptimo es de 5 a 8 personas, ya que un tamaño de equipo mayor producirá muchos puntos de fricción entre los miembros del equipo.

El tamaño del equipo Scrum debe ser manejable y adaptable y no se hará daño al equipo si durante el desarrollo del proyecto se adicionan nuevos integrantes. Este proceso de identificación debe comenzar temprano en el ciclo de vida del proyecto y el individuo seleccionado debe ser notificado de sus posibles funciones tan pronto como sea posible.

Un sello distintivo de equipos Scrum es el hecho que es dirigido sin el uso de la autoridad formal, por lo tanto al administrar equipos de Scrum, nunca es una buena idea decir esto es lo que quiero que hagan, ahora vayan a hacerlo! Utilice fórmulas de autorrealización de Maslow para que los miembros del equipo

usen lluvia de ideas para llegar a conceptos viables por ellos mismos. Este enfoque logrará en la práctica hacer que el equipo Scrum se sienta empoderado.

El éxito del equipo sólo puede garantizarse si cada uno conoce las reglas, y todos están de acuerdo en seguirlas.

Así como la claridad de las reglas y normas del equipo son importantes, también lo es la claridad de propósitos y objetivos. Los Daily Scrum se deben utilizar como un foro para reiterar el propósito y los objetivos del equipo. Cuanto más oyen hablar de él, más motivados estarán para lograr esos objetivos. [44]

[44] Jefferson Hanley (2015), Scrum: Your Quick Start Guide To Adopting Scrum For Your Organization.

6.7.2 Consejos Scrum exitosos: Según lo dice Hanley [45] Los profesionales de Scrum que lo han visto en acción le dirán que todo esto es más fácil decirlo que hacerlo, sin embargo, esos mismos profesionales también dirán que es posible ejecutar proyectos exitosos usando la metodología Scrum. Estos son algunos consejos prácticos que los profesionales de Scrum dan a los principiantes:

Obtener Educación: Si esta es su primera vez usando la metodología Scrum, no intente usarla sin antes evaluarse a si mismo; al equipo que desee en Scrum, debe tener el conocimiento suficiente en la metodología Scrum para implementarla con éxito.

Obtener el promotor adecuado: Scrum es diferente a las metodologías de gestión de proyectos tradicionales y es probable que haya más personas dentro de la organización que se oponen a su adopción. Sin embargo, si hay un promotor que está arriba en la jerarquía de la organización o que tenga una buena credibilidad entre los empleados de alto rango, las posibilidades de éxito de Scrum mejorará drásticamente.

En lugar de iniciar Scrum con el proyecto más difícil y desafiante primero debe utilizar algunos proyectos relativamente simples y factibles como casos de prueba de Scrum. Este enfoque dará al equipo la experiencia necesaria en el campo para abordar proyectos más complejos.

Ser realista: El pronóstico siempre es un asunto delicado cuando se trata de situaciones de la vida real. La mayoría de los nuevos practicantes de Scrum están demasiado entusiastas y en su afán de demostrar qué tan rápido puede entregar el producto con Scrum, subestiman los plazos de entrega. Siempre es una buena práctica agregar algunos tiempos extra a sus previsiones para que sean más realistas.

[45] Jefferson Hanley (2015), Scrum: Your Quick Start Guide To Adopting Scrum For Your Organization.

Iniciar con baja tecnología: No hay ninguna necesidad de apresurarse en salir y gastar una fortuna inicialmente para adquirir software de gestión de proyectos con Scrum. Incluso expertos profesionales en Scrum recomiendan el uso de pizarras o tableros con notas adhesivas para el control de eventos de Scrum.

Celebre el éxito: Una de las formas más importantes que uno puede poner en práctica la teoría de Maslow es celebrando el éxito del equipo. Las necesidades de autoestima y auto realización puede ser estimulado en gran medida cuando reciben elogios, premios y el reconocimiento por sus contribuciones. Eso es lo que lleva más lejos la consolidación de un equipo eficaz.

6.7.2.1 Afinando los consejos: Los consejos proporcionados están destinados a servir como orientaciones para poner Scrum en acción rápidamente y eficientemente. Sin embargo, uno debe darse cuenta de que las restricciones organizacionales individuales obstaculizan la adopción de algunos de esos consejos total o parcialmente.

Por ejemplo, la inversión en educación es una de las decisiones más importantes que tendrá que hacer una organización para preparar la adopción de Scrum. En el caso de las limitaciones presupuestarias, una buena opción es pagar para que una persona se capacite y esta se encargue de multiplicar este conocimiento al entrenar y capacitar al resto del equipo Scrum.

6.8 SCRUM PROFESSIONAL

Educarse acerca de los diversos aspectos de Scrum es una buena manera de asegurar que sus proyectos Scrum terminen con éxito. Sin embargo, no todos los miembros del equipo podrían estar interesado en la búsqueda de la formación a nivel de certificación de Scrum. Para ellos, tal vez asistir a talleres uno o dos días de Scrum puede ser un buen inicio.

Como bien lo dice Hanley [46] para los miembros del equipo que estén interesados en ir más allá con su formación Scrum, hay una serie de opciones de certificación disponibles según lo propuesto por la Alianza Scrum (Scrum alliance). Aquí están algunos a considerar:

Figura 17. Certificados Professional Scrum



Fuente: <https://www.scrumalliance.org/certifications>

[46] Jefferson Hanley (2015), Scrum: Your Quick Start Guide To Adopting Scrum For Your Organization.

6.8.1 Certificado Scrum master (CSM): Un Certificado de Scrum Master sirve como un recurso vital para ayudar a los equipos de Scrum, mejorando las posibilidades de éxito de un proyecto. Como CSM, se ganará una comprensión profunda de los valores y prácticas de Scrum y su aplicación va a proporcionar la experiencia a los equipos Scrum y el conocimiento que los manejadores de proyectos típicos no suelen tener.

Como un CSM, aprenderá cómo actuar como un líder, que le da el poder para dirigir el equipo sin autoridad formal sobre ellos. También dominará los principios de la estructura Scrum y aprenderá a ayudar al equipo a navegar en ellos. Lo más importante, se ganará habilidades valiosas sobre cómo aislar el equipo Scrum de distracciones internas y externas que a menudo sirven para descarrilar proyectos.

6.8.2 Certificado Scrum Product owner (CSPO): El papel del Product owner es una posición clave en el equipo Scrum. Como CSPO, usted va a adquirir un conocimiento profundo de todas las terminologías necesarias, principios y prácticas que le permitan cumplir con éxito sus responsabilidades como Scrum Product Owner. Como se ha señalado anteriormente, Product Owner es equiparable a Gerente de producto convencional, así como Gerente de Proyectos. La designación CSPO será de gran valor para cualquier persona deseosa de entrar en ese papel desafiante.

6.8.3 Certificado Scrum Developer (CSD): El curso CSD proporcionará a los desarrolladores de todo el conocimiento para mejorar sus habilidades de desarrollo ágil. Además, los desarrolladores también dominarán la ciencia detrás del desarrollo incremental como defiende Scrum, en lugar de la entrega efectuada en el enfoque del ciclo de vida de fin de proyecto. El CSD tiene una

ventaja sobre sus colegas no Scrum en que no sólo aprenden ingeniería Ágil, sino también están expuestos a los principios y prácticas del estructura Scrum básicos.

6.8.4 Certificado Scrum Professional (CSP): CSP son en la práctica los CSM, CSPO o CSD que deseen llevar su certificación Scrum al siguiente nivel. Cada metodología de entrega de proyectos se puede estirar hasta el límite y es entonces cuando las organizaciones ven beneficios adicionales. Esta formación le permitirá adquirir habilidades y conocimientos adicionales para ayudarle a desafiar a sus equipos Scrum para ampliar sus límites actuales en la práctica de esta metodología.

6.8.5 Certificado Scrum trainer (CST): Cuando la práctica de Scrum lo ha hecho perfecto en el campo de la ciencia del Scrum, es el momento de aprender el arte de enseñar Scrum. Como un CST, usted aprenderá todo lo que hay que saber sobre cómo traducir la riqueza de conocimientos y experiencia Scrum y transmitirla a los demás. Toda organización que está comprometida con Scrum debe considerar tener por lo menos un CST a bordo. El CST no sólo ayudará a los profesionales de Scrum en la organización a mantener sus conocimientos al día, a través de la formación continua, también podría llevar el rol de Scrum Master o Product owner, si es necesario.

6.8.6 Certificado Scrum Coach (CSC): Para los que desean elevar sus credenciales de Scrum, CSC podría ser la respuesta. Se debe ser capaz de demostrar que tiene todo el conocimiento práctico y teórico de Scrum para calificar como entrenador de otras personas, grupos y organizaciones. Un prerrequisito para alcanzar la designación CSC es que usted debe ser capaz de

demostrar que ha ayudado al menos una organización a adoptar con éxito Scrum en la ejecución de proyectos de la vida real.

Gráfico de cómo le aporta a cada miembro del equipo Scrum cada una de las certificaciones:

Figura 18. Roles de Scrum frente a las certificaciones

Pre-Agile Role	Agile Role	CSM	CSPO	CSD
Programmer	Team Member	●		●
Tester	Team Member	●		●
Team Lead	Team Member or ScrumMaster	●	●	
Project Manager	ScrumMaster	●	●	
Product Manager	Product Owner	●	●	
Manager	Manager	●		
Business Analyst	Team Member		●	

● Core
● Recommended
● Optional

Fuente: <https://www.scrumalliance.org/certifications/practitioners>

7. CONSULTA A TRAVÉS DE ENCUESTA A ESTUDIANTES DE SEXTO SEMESTRE EN ADELANTE DEL PROGRAMA DE INGENIERÍA DE SISTEMAS Y COMPUTACIÓN DE LA UNIVERSIDAD TECNOLÓGICA DE PEREIRA ALGUNAS PREGUNTAS QUE PERMITEN CONCLUIR LA NECESIDAD DE INCLUIR AL PENSUM LA METODOLOGÍA ÁGIL SCRUM PARA MEJORAR EL TRABAJO EN EQUIPO.

El presente capítulo se presentan dos encuestas: Estudiantes activos y Egresados.

7.1 RESUMEN ENCUESTA DE ESTUDIANTES ACTIVOS

FICHA TÉCNICA DE LA ENCUESTA

Diseño muestral	Probabilístico
Población objetivo	Estudiantes activos de ISC de sexto semestre en adelante de jornada diurna y nocturna
Universo representado	358 estudiantes del día y la noche de sexto semestre en adelante de jornada diurna y nocturna
Técnica	Encuesta electrónica dirigida persona a persona
Tamaño de muestra	44 estudiantes activos que representan un 12.3% del universo
Momento estadístico	Febrero 1 al 4 de 2015
Financiación	Recursos propios
Margen de error	14% error estándar relativo máximo
Heterogeneidad	50% (Lo habitual suele ser 50%)
Nivel de confianza	95% (Lo habitual suele ser entre el 95% y el 99%)

Ver toda la encuesta con resultados y graficas Anexo A

7.1.1 Análisis de la presente encuesta:

¿Le gusta programar?

80% le gusta programar y 20% no le gusta.

¿Cómo le fue en las asignaturas de programación?
--

31.11% muy bien, 44.44% bien, 17.77% regular, 6.68% mal.
--

¿Cómo cree que se hace un desarrollo de Software en equipo?

22.22% con métodos ágiles, 77.78% no sabe una manera adecuada de hacerlo.

¿Alguna vez ha escuchado de metodologías Ágiles?
--

53.34% dice haber escuchado alguna vez de metodologías ágiles y el 46.66% no la ha escuchado nunca.

¿Alguna vez ha escuchado de metodologías Ágiles?
--

53.34% dice haber escuchado alguna vez de metodologías ágiles y el 46.66% no la ha escuchado nunca.

¿Qué tanto conoce de la metodología Scrum?
--

26.66% conoce perfectamente la metodología Scrum y los
--

demás conocen muy poco de ella o solo han escuchado algo.

¿En qué le gustaría ejercer su profesión?

Al 75.55% Desarrollo de software, 42.22% en Gerencia de Proyectos de Ingeniería, 42.22% en Gerencia de proyectos de software, 40% en Crear empresa en desarrollo tecnológico, 37.77% como Analista de sistemas, 33.33% en Consultoría y asesoría en Informática, Administración en servicios Informáticos y comunicaciones, Investigación en ciencias de la computación, 31.11% en Gerencia de Proyectos de información, 26% en Desarrollar proyectos de redes de comunicaciones.

Con todos estos valores porcentuales se puede concluir claramente la necesidad de incluir al pensum la metodología ágil Scrum que es la más utilizada actualmente en la mayoría de empresas desarrolladoras de software en Colombia y el mundo (ver figura 1), dicha inclusión permitirá mejorar estos indicadores pues hará que los estudiantes puedan optimizar el trabajo en equipo en desarrollo de software.

Otra conclusión importante encontrada en esta encuesta es que los dos lenguajes preferidos son Java con el 71.11% y Python con el 66.66% que son igualmente dos de los lenguajes más utilizados en el mundo Según Tiobe [47] al igual que JavaScript y estos tres lenguajes poco se ven en el programa, siendo C el que más se utiliza.

[47] TIOBE, Top 20 Programming Languages, <http://www.tiobe.com/index.php/content/paperinfo/tpci/index.html>

El resultado anterior se debe interpretar así:

La encuesta fue de 44 personas de un universo de 358, el 95% de las veces el dato que se quiere medir estará en el intervalo $\pm 14\%$ respecto al dato que se observó en la encuesta.

Anexo A. Encuesta de estudiantes activos de Ingeniería de Sistemas

7.2 RESUMEN ENCUESTA DE EGRESADOS

FICHA TÉCNICA DE LA ENCUESTA [48]

Diseño muestral	Probabilístico
Población objetivo	Egresados de ISC desde 2010 a 2014
Universo representado	250 egresados de 2010 a 2014
Técnica	Encuesta electrónica dirigida persona a persona
Tamaño de muestra	10 egresados que representa el 4% del universo
Momento estadístico	Febrero 1 al 4 de 2015
Financiación	Recursos propios
Margen de error	32% error estándar relativo máximo
Heterogeneidad	50% (Lo habitual suele ser 50%)
Nivel de confianza	95% (Lo habitual suele ser entre el 95% y el 99%)

[48] NETQUEST, <http://www.netquest.com/es/panel/calculadora-muestras.html>

Ver toda la encuesta con resultados y graficas Anexo B

Ver toda la encuesta con resultados y graficas Anexo A

7.2.1 ANÁLISIS DE LA PRESENTE ENCUESTA:

¿Le gusta programar?

80% le gusta programar y 20% no le gusta.

¿Cómo le fue en las asignaturas de programación?

20% muy bien, al 60% bien, al 10% regular, 10% mal.

¿Cómo cree que se hace un desarrollo de Software en equipo?

30% con métodos ágiles, 50% no sabe una manera adecuada de hacerlo, 20% con PSP que tampoco se enseña en ISC-UTP

¿Alguna vez ha escuchado de metodologías Ágiles?

40% dice haber escuchado alguna vez de metodologías ágiles y el 60% no la ha escuchado nunca.

¿Qué tanto conoce de la metodología Scrum?

40% conoce Scrum medianamente, 20% muy poco, 10% no conoce nada y 10% lo conoce perfectamente.

¿En qué le gustaría ejercer su profesión?

Al 50% Desarrollo de software, 40% en Gerencia de Proyectos de software y Crear empresa en desarrollo tecnológico, 30% en Gerencia de proyectos de Ingeniería y Analista de sistemas, 20% en Administración en servicios Informáticos y comunicaciones e Investigación en ciencias de la computación, 10% en Gerencia de Proyectos de información y en Desarrollar proyectos de redes de comunicaciones, 0% en Consultoría y asesoría en Informática.

Con todos estos valores porcentuales se puede concluir claramente cómo se dio en el análisis anterior de los estudiantes activos, la necesidad de incluir al pensum la metodología ágil Scrum, dicha inclusión permitirá mejorar estos indicadores pues hará que los egresados tengan conocimiento y práctica en trabajo en equipo en desarrollo de software en una metodología altamente usado en la industria como Scrum. (Ver figura 1)

Otra conclusión importante encontrada en esta encuesta es que los dos lenguajes preferidos son C++ y C# con el 50% de preferencia seguidos por Java y Python con el 40% que son igualmente otros de los lenguajes más utilizados en el mundo Según Tiobe [49] al igual que JavaScript y estos cinco lenguajes poco o nada se ven en el programa, siendo C el que más se utiliza.

El resultado anterior se debe interpretar así:

La encuesta fue de 10 personas de un universo de 250, el 95% de las veces el dato que se quiere medir estará en el intervalo $\pm 32\%$ respecto al dato que se observó en la encuesta.

Anexo B. Encuesta de estudiantes egresados de Ingeniería de Sistemas

[49] TIOBE, Top 20 Programming Languages, <http://www.tiobe.com/index.php/content/paperinfo/tpci/index.html>

7.3 ANÁLISIS DE LA COMPARACIÓN DE LAS DOS ENCUESTAS:

	¿Le gusta programar?
Estudiantes	80% le gusta programar y 20% no le gusta.
Egresados	80% le gusta programar y 20% no le gusta.

Tanto estudiantes como egresados tienen el mismo porcentaje de gusto por la programación con un 80%, lo que hace ver la preferencia de los egresados y futuros egresados trabajar en la industria de desarrollo de Software.

	¿Cómo le fue en las asignaturas de programación?
Estudiantes	31.11% muy bien, 44.44% bien, 17.77% regular, 6.68% mal.
Egresados	20% muy bien, al 60% bien, al 10% regular, 10% mal.

A los estudiantes y egresados les fue bien en las asignaturas de programación, el 75.55% de los estudiantes y el 80% de los egresados lo que hace ver que además del gusto por la programación de la pregunta anterior se puede apreciar en esta el rendimiento que se refleja en las notas obtenida en esta área.

	¿Cómo cree que se hace un desarrollo de Software en equipo?
Estudiantes	22.22% con métodos ágiles, 77.78% no sabe una manera adecuada de hacerlo.
Egresados	30% con métodos ágiles, 50% no sabe una manera adecuada de hacerlo, 20% con PSP que tampoco se enseña en ISC-UTP

Se observa que tanto estudiantes como egresados tienen un muy bajo conocimiento de metodologías ágiles para desarrollo de software en equipo, lo que hace ver que dentro del pensum del programa es muy escaso la orientación de una metodología como Scrum.

	¿Alguna vez ha escuchado de metodologías Ágiles?
Estudiantes	53.34% dice haber escuchado alguna vez de metodologías ágiles y el 46.66% no la ha escuchado nunca.
Egresados	40% dice haber escuchado alguna vez de metodologías ágiles y el 60% no la ha escuchado nunca.

Los egresados y estudiantes dicen haber escuchado entre el 40% y el 53.34% respectivamente alguna vez sobre metodologías ágiles, pero se puede denotar que solo ese porcentaje lo ha escuchado nombrar ya que en la pregunta anterior se puede apreciar que los porcentaje de conocimiento en Scrum son muy inferiores ya que solo entre el 22.22% y el 30% cree que el desarrollo de software en equipo se debe hacer con una metodología ágil.

	¿Qué tanto conoce de la metodología Scrum?
Estudiantes	26.66% conoce perfectamente la metodología Scrum y los demás conocen muy poco de ella o solo han escuchado algo.
Egresados	40% conoce Scrum medianamente, 20% muy poco, 10% no conoce nada y 10% lo conoce perfectamente.

Con esta pregunta se confirma el poco conocimiento de la metodología Scrum ya que el 26.66% de los estudiantes dice conocer perfectamente la metodología Scrum y solo un 10% de los egresados dice hacerlo lo que ratifica el muy poco conocimiento de la metodología Scrum a todo nivel tanto de estudiantes como en egresados.

	En que le gustaría ejercer su profesión?
Estudiantes	Al 75.55% Desarrollo de software, 42.22% en Gerencia de Proyectos de Ingeniería, 42.22% en Gerencia de proyectos de software, 40% en Crear empresa en desarrollo tecnológico, 37.77% como Analista de sistemas, 33.33% en Consultoría y asesoría en Informática, Administración en servicios Informáticos y comunicaciones, Investigación en ciencias de la computación, 31.11% en Gerencia de Proyectos de información, 26% en Desarrollar proyectos de redes de comunicaciones.
Egresados	Al 50% Desarrollo de software, 40% en Gerencia de Proyectos de software y Crear empresa en desarrollo tecnológico, 30% en Gerencia de proyectos de Ingeniería y Analista de sistemas, 20% en Administración en servicios Informáticos y comunicaciones e Investigación en ciencias de la computación, 10% en Gerencia de Proyectos de información y en Desarrollar proyectos de redes de comunicaciones, 0% en Consultoría y asesoría en Informática.

En esta pregunta que tiene que ver de en donde desearía trabajar para ejercer su profesión, se puede apreciar que el mayor porcentaje tanto en estudiantes como en egresados se inclina por querer trabajar en la industria de desarrollo de Software.

7.4 CONCLUSIÓN FINAL DEL PRESENTE CAPÍTULO:

Los estudiantes y egresados del programa tiene un alto grado de gusto por la programación (80%) y además tuvieron un buen desempeño en las asignaturas correspondientes (Estudiantes 75.55% y egresados 80%), a la pregunta dónde desearía trabajar para ejercer su profesión, los estudiante y egresados del programa de ISC-UTP se inclinan mayoritariamente por querer trabajar en la industria de desarrollo de Software (Estudiantes 75.55% y egresados 50%).

Tanto estudiantes como egresados tienen un muy bajo conocimiento de metodologías ágiles para desarrollo de software en equipo (Estudiantes 22.22% y egresados 30%), lo que hace ver que dentro del pensum del programa es muy escaso la orientación de una metodología como Scrum.

Los egresados y estudiantes dicen haber escuchado entre el 40% y el 53.34% respectivamente alguna vez sobre metodologías ágiles, pero se puede denotar que solo ese porcentaje lo ha escuchado nombrar ya que se puede apreciar que los porcentaje de conocimiento de Scrum para hacer desarrollo de Software en equipo son muy inferiores (Estudiantes 22.22% y egresados 30%)

¿Qué tanto conoce de la metodología Scrum? Con esta pregunta se confirma el poco conocimiento de la metodología Scrum ya que el 26.66% de los estudiantes dice conocer perfectamente la metodología Scrum y solo un 10% de los egresados dice hacerlo, lo que ratifica el muy poco conocimiento de la metodología Scrum a todo nivel tanto de estudiantes como en egresados.

Lo que permite concluir la inmensa necesidad de incluir dentro del pensum de ISC-UTP una metodología ágil como Scrum.

8. ENTREVISTA CON DOCENTES DE INGENIERÍA DE SISTEMAS Y COMPUTACIÓN DE LA NECESIDAD DE INCLUIR AL PENSUM LA METODOLOGÍA SCRUM.

FICHA TÉCNICA DE LA ENTREVISTA

Diseño muestral	Probabilístico
Población objetivo	Docentes de áreas afines al estudio de ISC
Universo representado	40 Docentes
Técnica	Entrevista cualitativa semiestructurada persona a persona
Tamaño de muestra	9 docentes que representan el 22.5% del universo
Momento estadístico	19 de Septiembre de 2014 hasta Febrero 6 de 2015
Financiación	Recursos propios
Margen de error	30% error estándar relativo máximo
Heterogeneidad	50% (Lo habitual suele ser 50%)
Nivel de confianza	95% (Lo habitual suele ser entre el 95% y el 99%)

En el presente capítulo se consulta por medio de entrevistas persona a persona con dos Ingenieros no egresados de la Universidad Tecnológica de Pereira que trabajan en dos empresas desarrolladoras de software en la región y con nueve docentes de Ingeniería de Sistemas de la Universidad, con los que se socializa desde el inicio la idea del proyecto actual para sugerir la necesidad de incluir al pensum la metodología Scrum, pero no de simplemente enunciar que es, sino que se profundice en ella de manera teórica y práctica, buscando con ello atacar el problema actual del desconocimiento de los estudiantes de la forma adecuada de hacer un trabajo en equipo en desarrollo de software, lo que hace que les genere una gran dificultad al entrar al mercado laboral, donde la gran mayoría de las empresas de Colombia y el mundo utilizan dicha metodología

para ser más eficientes en cuanto a tiempos y costos en todos los desarrollos de software.

Se les pide la opinión a ellos sobre el proyecto y lo que piensan de sí es buena o mala idea de incluirlo en el programa, estos son sus opiniones:

El ingeniero Juan Carlos Mejía Director Ejecutivo de **JAiVAna** Soluciones Empresariales dijo que como propone la metodología Scrum el trabajo colaborativo en un proyecto de desarrollo de Software era una excelente metodología pero algunas empresas que trabajan con esta filosofía la han “maltratado” mucho, con esto se refiere a que usan procesos Scrum combinándolas con otros tipos de procesos ajenos a esta metodología. Con respecto al proyecto, le parece que es una muy buena iniciativa por parte de nosotros el hacer la propuesta de inclusión al pensum algo tan valioso como usar una metodología ágil para desarrollo de Software de Calidad adoptando esta filosofía desde tempranos semestres de formación como Ingeniero de Sistemas y Computación.

El Ingeniero John Alexander Holguín trabaja actualmente en Geminus Software de Colombia S.A.S. utilizando metodología Scrum, dijo que trabajar colaborativamente es la mejor manera de sacarle provecho a los talentos que tienen cada uno de los que hacen parte del grupo de trabajo y que mejor forma que hacer esto con una metodología ágil de desarrollo de Software. Desde hace varios años ha trabajado con Scrum haciendo que la empresa sea más productiva y algo muy importante que es tener al cliente satisfecho con los productos que se desarrollan. Le parece que la propuesta de inclusión al pensum la enseñanza de la metodología Scrum es muy bueno e importante para la formación de ingenieros de Sistemas.

El ingeniero Jorge Alberto Gálvez docente de la asignatura Ingeniería de Software I y II dijo que hace varios semestres atrás han ido tratando de implementar nuevos contenidos al pensum del programa de Ingeniería de Sistemas y Computación y que esta propuesta sale en un buen momento. De acuerdo a lo que se ha hablado en las reuniones para definir el contenido de las asignaturas es enseñar desarrollo de Software en la asignatura Ingeniería del Software II y enseñar en otra materia los contenidos administrativos al respecto. Sugirió que la propuesta de inclusión de la metodología Scrum se haga en una asignatura administrativa como Administración de Proyectos de Software ya que los contenidos se pueden acoplar fácilmente.

El ingeniero Carlos Mario Medina docente de la asignatura Arquitectura Cliente/Servidor y computación gráfica dijo que le parecía una propuesta muy interesante ya que considera que los estudiantes de Ingeniería de Sistemas y Computación deben saber desarrollar proyectos de Software en equipo, teniendo en cuenta que las asignaturas de programación son muy importantes para el desarrollo del pensamiento lógico.

La Ingeniera Luz Stella Valencia docente de gerencia de proyectos, emprendimiento, administración de empresas e investigación de operaciones, dijo que la propuesta le parecía muy importante que se implementara ya que muchas empresas del sector de desarrollo de Software han adoptado este tipo de metodologías para la administración de proyectos de desarrollo en gran porcentaje Scrum. “Todos quieren con Scrum” ya que la filosofía que maneja ayuda a mejorar los procesos internos de Administración de proyectos de desarrollo de Software.

El ingeniero Guillermo González docente de administración de proyectos de software dijo que la enseñanza de metodologías ágiles como Scrum

enriquecería mucho la vida profesional de un ingeniero de Sistemas ya que la optimización de procesos es uno de los pilares de esta carrera profesional y Scrum ayuda a mejorar estos procesos con el Scrum Master de líder. Le parece acertada la propuesta de inclusión de esta metodología al pensum del programa de Ingeniería de Sistemas y Computación para la enseñanza a los ingenieros de Sistemas.

El ingeniero Wilman Tello docente en el semestre 2014-2 de Ingeniería de software dijo que le parece una muy buena idea la inclusión de Scrum en el programa ya que el, en laboratorio de software explica un poco que es Scrum pero se enfoca más a la parte práctica, pero los estudiantes teniendo toda la teoría de la metodología y hacer la práctica en Laboratorio es lo ideal ya que permite que un desarrollo de software se haga en equipo de una manera incremental y el Scrum master pueda hacer el seguimiento y control de cada uno de las tareas de cada rol, dividido en sprint de una semana cada uno.

El ingeniero Oscar Hernando Marín docente en el semestre 2014-2 de Planeación estratégica de sistemas dijo que le parece una buena idea pues no solo deben ser buenos programadores sino que deben saber trabajar en equipo de una manera eficiente con una metodología ágil.

El ingeniero Juan de Jesús Veloza docente de Administración de sistemas de información, sistemas operativos I, administración de proyectos de software, dijo que este semestre tiene previsto dar una teoría de Scrum en la segunda semana de clase en administración de proyectos de software y en la tercera semana dará la teoría de PSP, se le informo el resultado de las encuestas con estudiantes donde se plasma el desconocimiento de metodologías de trabajo en equipo para desarrollo de software en los estudiantes activos y egresados y la

necesidad que Scrum no solo sea de manera teórica y en una solo semana sino hacerla también de manera práctica y de una manera más profunda para que los estudiantes interioricen la manera correcta de trabajar en equipo en dicha área.

El ingeniero Alejandro Rodas docente de programación III y programación IV, dijo que es una excelente idea enseñar Scrum en el programa pero no solo enseñarlo sino ponerlo en práctica e implementarlo para que los estudiantes aprendan muy bien a trabajar en equipo para hacer desarrollo de software, hacer toma de requerimientos, análisis y haciendo control de versiones de manera incremental.

El ingeniero Carlos Alberto Ocampo docente de introducción a la informática, base de datos I, proyecto de grado I e ingeniería de software III dijo que le parece una muy buena idea que se implemente Scrum en el programa desde sexto semestre y que se haga práctico en Laboratorio de software, ya que es una necesidad que se ha visto reflejada en *Indra* donde hay casi cincuenta ingenieros de la universidad y ellos han tenido el mismo problema que se vio reflejado en nuestra encuesta de egresados, la falta de conocimiento de metodologías ágiles que optimicen el trabajo en equipo. También habló que Mintic está invirtiendo miles de millones de pesos para capacitar a profesionales TI con PSP y quienes más invierten en esa área son los chinos y los japoneses ya que los hace más competitivos.

8.1 ANÁLISIS DE LA PRESENTE ENTREVISTA

A la conclusión que se llega después de las entrevistas con los docentes es una amplia aceptación a la propuesta de la metodología ágil Scrum en el programa de ingeniería de sistemas y computación ya que se debe enseñar no solamente de manera teórica sino práctica es la forma más idónea para que los estudiantes aprendan a trabajar en equipo en desarrollo de software y así logren tener una entrada al campo laboral usando una metodología actual y ampliamente usada como Scrum. Además de todas las consideraciones anteriores, otra de las razones importantes para sustentar la relevancia de la inclusión de Scrum dentro del programa ISC-UTP es que MINTIC desde el 2010 viene haciendo una gran inversión en la industria TI de Colombia buscando a corto plazo hacer de Colombia una potencia mundial en desarrollo de Software proyectando tener una facturación anual de 18 billones de pesos en el año 2018, MINTIC ofrece créditos condonables hasta del 100% a través del ICETEX con diferentes certificaciones, entre ellos las certificaciones [50] de Scrum Master, Scrum Product Owner.

[50] CRÉDITOS CONDONABLES PARA LA CERTIFICACIÓN EN COMPETENCIAS ESPECÍFICAS Y TRANSVERSALES

<https://www.icetex.gov.co/dnnpro5/Portals/0/Documentos/Fondos/2aConvCompetenciasTransversales15ene2015.pdf>

9. REDACCIÓN DE LA PROPUESTA DE INCLUSIÓN DE LA METODOLOGÍA SCRUM AL PENSUM DE ISC-UTP HACIENDO VER SU IMPORTANCIA PARA LOS ESTUDIANTES DEL PROGRAMA.

Teniendo presente las tres consideraciones que se resumen en el presente capítulo, que se vieron en una forma más amplia en los capítulos seis, siete y ocho se propone poner en consideración al programa de Ingeniería de Sistemas y Computación de la Universidad Tecnológica de Pereira, incluir en el pensum la enseñanza de la metodología ágil Scrum de una manera amplia en la asignatura que el programa considere pertinente y la parte práctica en laboratorio de software.

9.1 LA ELECCIÓN DE SCRUM EN VEZ DE METODOLOGÍAS DE GESTIÓN DE PROYECTOS CONVENCIONALES PROPORCIONA MEJORES BENEFICIOS, TALES COMO:

Un aumento en el retorno de inversión de las organizaciones mediante la producción de un conjunto estable y continuo de los beneficios de valor agregado en lugar de la entrega de un producto al final de un proyecto.

Dado que todos los actores están continuamente y constantemente ocupados con el proyecto a través de Sprints exitosos, el resultado es más confiable que una que se relaciona con las partes interesadas en la reunión de toma de requisitos al inicio y al final con las pruebas de aceptación.

El riesgo general del proyecto y la incertidumbre se reduce drásticamente como resultado de descomponer los entregables en fracciones más pequeñas y casi autónomas.

Mediante la producción del entregable al final de los Sprint, el tiempo de mercado de la organización aumenta porque los usuarios no tienen que esperar hasta que el producto esté totalmente hecho antes de la ejecución.

Un equipo Scrum bien organizado es un grupo ágil y eficiente que ofrece un trabajo productivo, creativo e innovador en la organización.

9.2 INFORMACIÓN OBTENIDA DE LAS ENCUESTAS REALIZADAS A LOS ESTUDIANTES ACTIVOS Y EGRESADOS DE INGENIERÍA DE SISTEMAS Y COMPUTACIÓN PARA LA INCLUSIÓN DE SCRUM EN EL PROGRAMA:

Los estudiantes y egresados del programa tiene un alto grado de gusto por la programación (80%) y además tuvieron un buen desempeño en las asignaturas correspondientes (Estudiantes 75.55% y egresados 80%), a la pregunta dónde desearía trabajar para ejercer su profesión, los estudiantes y egresados del programa de ISC-UTP se inclinan mayoritariamente por querer trabajar en la industria de desarrollo de Software (Estudiantes 75.55% y egresados 50%).

Tanto estudiantes como egresados tienen un muy bajo conocimiento de metodologías ágiles para desarrollo de software en equipo (Estudiantes 22.22% y egresados 30%), lo que hace ver que dentro del pensum del programa es escasa la orientación de metodologías que permiten trabajar eficientemente en equipo al hacer desarrollo de software.

Los egresados y estudiantes dicen haber escuchado entre el 40% y el 53.34% respectivamente alguna vez sobre metodologías ágiles, pero se puede denotar que solo ese porcentaje lo ha escuchado nombrar ya que se puede apreciar que los porcentaje de conocimiento de Scrum para hacer desarrollo de Software en equipo son muy inferiores (Estudiantes 22.22% y egresados 30%)

¿Qué tanto conoce de la metodología Scrum? Con esta pregunta se confirma el poco conocimiento de la metodología Scrum ya que el 26.66% de los estudiantes dice conocer perfectamente la metodología Scrum y solo un 10% de los egresados dice hacerlo, lo que ratifica el poco conocimiento de la metodología Scrum a todo nivel tanto en estudiantes como en egresados.

9.3 INFORMACIÓN OBTENIDA EN LA ENTREVISTA CON DOCENTES DE ISC-UTP AFINES A INGENIERÍA DE SOFTWARE Y LA PROMOCIÓN DE MINTIC EN EL SECTOR DE DESARROLLO DE SOFTWARE:

Hay una amplia aceptación a la propuesta de la metodología ágil Scrum en el programa de ingeniería de sistemas y computación por parte de los docentes al considerar que es una muy buena metodología para que aprendan los estudiantes, se debe enseñar no solamente de manera teórica sino práctica, es la forma más idónea para que los estudiantes aprendan a trabajar en equipo en desarrollo de software y así logren tener una entrada al campo laboral usando una metodología actual y ampliamente usada como Scrum.

Otra de las razones importantes para soportar la relevancia de la inclusión de Scrum dentro del programa ISC-UTP es estar en sintonía con todo el esfuerzo que está realizando MINTIC desde el 2010, donde viene haciendo una gran inversión en la industria TI nacional, buscando a corto plazo hacer de Colombia una potencia mundial en desarrollo de Software, proyectando tener una facturación anual de 18 billones de pesos en el año 2018, para lograr esta meta, MINTIC ofrece créditos condonables hasta del 100% a través del ICETEX con diferentes certificaciones entre ellos las certificaciones de Scrum Master, Scrum Product Owner. [51]

[51] CRÉDITOS CONDONABLES PARA LA CERTIFICACIÓN EN COMPETENCIAS ESPECÍFICAS Y TRANSVERSALES

<https://www.icetex.gov.co/dnnpro5/Portals/0/Documentos/Fondos/2aConvCompetenciasTransversales15ene2015.pdf>

9.4 CONCLUSIÓN DEL CAPITULO:

Teniendo presente los tres puntos anteriores donde se enuncia las ventajas de la metodología Scrum frente a metodologías de gestión de proyectos convencionales, la necesidad de aprovechar el gusto y el desempeño por la programación de los estudiantes y egresados de ISC-UTP, así como también la manifiesta intención de ellos de querer trabajar en desarrollo de proyectos de software; además del desconocimiento de la metodología Scrum para hacer un buen trabajo en equipo en dicha área.

Incluir Scrum dentro del programa de ISC-UTP mejorará el desempeño de los estudiantes en desarrollo de software en equipo y además los pondrá en sintonía con una política de gobierno que lidera MINTIC desde el año 2010 también les permitirá a los egresados tener una mejor competencia al querer ingresar al mercado laboral en desarrollo de proyectos de software.

10. CONCLUSIONES

Estas son las principales conclusiones del presente proyecto de grado en la Universidad Tecnológica de Pereira en el programa de Ingeniería de Sistemas y Computación:

10.1 LOS ESTUDIANTES OBTIENEN MAYORES COMPETENCIAS APRENDIENDO SCRUM SI ES INCLUIDO EN EL PROGRAMA TALES COMO:

Aprender a trabajar en equipo donde cada uno cumple un rol específico con objetivos divididos en Sprint.

El riesgo general del proyecto y la incertidumbre se reduce drásticamente además de un aumento en el retorno de inversión como resultado de descomponer los entregables en fracciones más pequeñas y casi autónomas en lugar de la entrega de un producto al final de un proyecto. Mediante la producción del entregable al final de los Sprint, el tiempo de mercado de la organización aumenta porque los usuarios no tienen que esperar hasta que el producto esté totalmente hecho antes de la ejecución.

10.2 CONCLUSIONES DE ENCUESTAS REALIZADAS A LOS ESTUDIANTES ACTIVOS Y EGRESADOS:

Los estudiantes y egresados del programa tienen un alto grado de gusto y buen desempeño por la programación, desean mayoritariamente trabajar en la industria de desarrollo de Software pero tienen un muy bajo conocimiento de metodologías para desarrollo de software en equipo.

10.3 CONCLUSIÓN DE LA ENTREVISTA CON DOCENTES Y LA PROMOCIÓN DE MINTIC EN EL SECTOR DE DESARROLLO DE SOFTWARE:

Hay amplia aceptación a la propuesta de la metodología ágil Scrum en el programa de ingeniería de sistemas y computación ya que se debe enseñar no solamente de manera teórica sino práctica, es la forma más idónea para que los estudiantes aprendan a trabajar en equipo en desarrollo de software y así logren tener una entrada al campo laboral usando una metodología vigente y ampliamente usada como Scrum.

Además de todas las consideraciones anteriores, otra de las razones importantes para sustentar la relevancia de la inclusión de Scrum dentro del programa ISC-UTP es que MINTIC desde el 2010 viene haciendo una gran inversión en la industria TI nacional, buscando a corto plazo hacer de Colombia una potencia mundial en desarrollo de Software, proyectando tener una facturación anual de 18 billones de pesos en el año 2018, MINTIC ofrece créditos condonables hasta del 100% a través del ICETEX con diferentes certificaciones entre ellos las certificaciones de Scrum Master, Scrum Product Owner.

Con el presente proyecto de grado se aspira lograr persuadir a la dirección del programa de ingeniería de sistemas y computación la importancia de incluir dentro del pensum del programa de una manera amplia y lograr que los estudiantes logren ser más eficientes en el trabajo en equipo en desarrollo de software.

11. BIBLIOGRAFÍA

Jefferson Hanley (2015), Scrum: Your Quick Start Guide To Adopting Scrum For Your Organization

CIRCULO CUADRADO, La Jerarquía de las necesidades, según Maslow,
<http://circulocuadrado.es/blog/la-motivacion-humana-2a-parte/>

DEEMER, Pete., Benefield, Gabrielle., Larman, Craig y Vodde, Bass. (2012). A Lightweight Guide to the Theory and Practice of Scrum. Recuperado de
<http://scrumfoundation.com/library>

TIOBE, Top 20 Programming Languages,
<http://www.tiobe.com/index.php/content/paperinfo/tpci/index.html>

GALLEGO Manuel Trigas, Gestión de proyectos informáticos, metodología Scrum.
<http://openaccess.uoc.edu/webapps/o2/bitstream/10609/17885/1/mtrigasTFC0612memoria.pdf>

HUNDERMARK, Peter. (2009). Un Mejor Scrum. Recuperado de
<http://www.scrumsense.com/>

HIGHSMITH Jim 2001, The Agile Manifesto.

<http://www.drdoobs.com/open-source/the-agile-manifesto/184414755>

INSTITUTO COLOMBIANO DE NORMAS TÉCNICAS, Normas Colombianas para la presentación de trabajos de investigación, sexta actualización, Bogotá D.C.: ICONTEC, 2008 NTC 1486. 41p.

<http://recursosbiblioteca.utp.edu.co/OVA2/word.htm>

INGENIERÍA DE SISTEMAS Y COMPUTACIÓN,

isc.utp.edu.co/informacion/pensum-ingenieria-de-sistemas-y-computacion-.html

INTERSOFTWARE, Certifíquese en SCRUM con un crédito 100% condonable,
<http://www.intersoftware.org.co/content/certifiquese-en-scrum-con-un-credito-100-condonable>

MINTIC, <http://www.mintic.gov.co/portal/604/w3-channel.html>

NETQUEST, <http://www.netquest.com/es/panel/calculadora-muestras.html>

Netquest.com SOLUCIONES NETQUEST DE INVESTIGACIÓN S.L.

<http://www.netquest.com/es/panel/calculadora-muestras.html>

PMI Northern Chapter, <http://projectmanager.org/>

PMI, Project Manage Institute,

<http://americalatina.pmi.org/latam/AboutUS/WhatIsPMI.aspx>

STATE of agile. (2014). VersionOne. <http://stateofagile.versionone.com>

Metodología en Cascada, www.gavide.com

Metodología Ágil Crystal Clear, <https://iswugaps2crystalclear.wordpress.com/>

Estructura Scrum,

http://projectmanager.org/content.php?page=Certified_Scrum_Product_Owner

BARIS DERE, Scrum, <http://www.barisdere.com/2010/12/cevik-surecler-agile-scrum/>

PITA Lab, <http://pitalab.com>

MSDN Blogs, Branching for Scrum,

<http://blogs.msdn.com/b/billheys/archive/2011/01/18/branching-for-scrum.aspx>

Origen del Product Backlog, <http://guntherverheyen.com/category/agile/my-fragility-agile/page/2/>

SCRUM Reference Card, <http://scrumreferencecard.com/scrum-reference-card/>

Desarrollo del Sprint backlog,
<http://www.methodsandtools.com/archive/atddreadysprintbacklog.php>

Burn down chart, http://www.scrum-institute.org/Burndown_Chart.php

SCRUM ALLIANCE, Certifications, <https://www.scrumalliance.org/certifications>

Roles de Scrum frente a las certificaciones,
<https://www.scrumalliance.org/certifications/practitioners>

ANEXO A

Encuesta de estudiantes activos de Ingeniería de Sistemas

Estudiantes de sexto semestre en adelante del programa de Ingeniería de Sistemas y Computación de la Universidad Tecnológica de Pereira, estos son los resultados de las tablas estadísticas logradas con esta investigación:

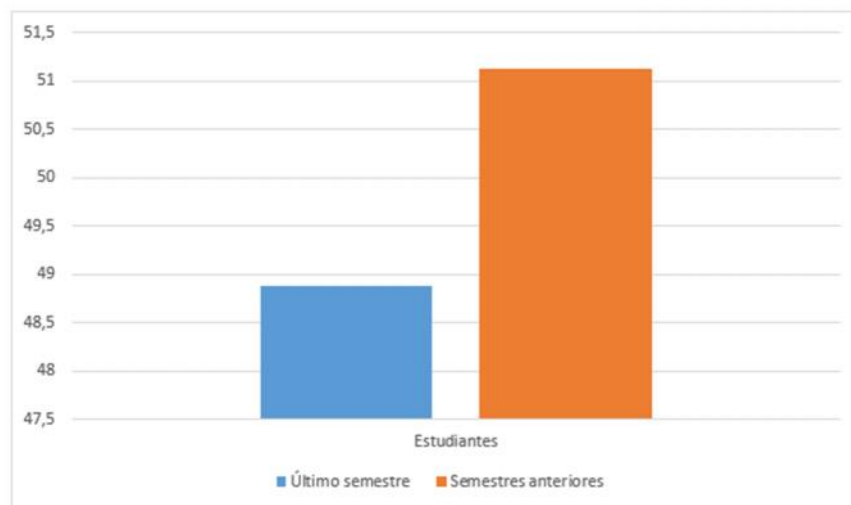
ENCUESTA DE ESTUDIANTES ACTIVOS INGENIERÍA DE SISTEMAS Y COMPUTACIÓN

Respondida por 44 estudiantes

a. ¿Cuántos semestres le faltan para terminar Ingeniería de Sistemas y Computación?

Solo este semestre	48.88%
Dos	24.44%
Tres	11.11%
Cuatro	13.33%
Cinco	2.24%
Seis	0%

El 48.88% de los estudiantes que respondieron esta encuesta son de último semestre, el 51.12% restante está en los últimos cinco semestres

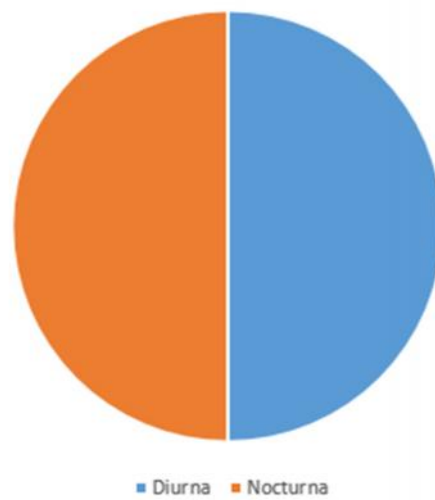


b. ¿A qué Jornada pertenece?

Diurna	51.11%
Jornada Especial	48.89%

En la encuesta hay igualdad de estudiantes de jornada diurna y nocturna.

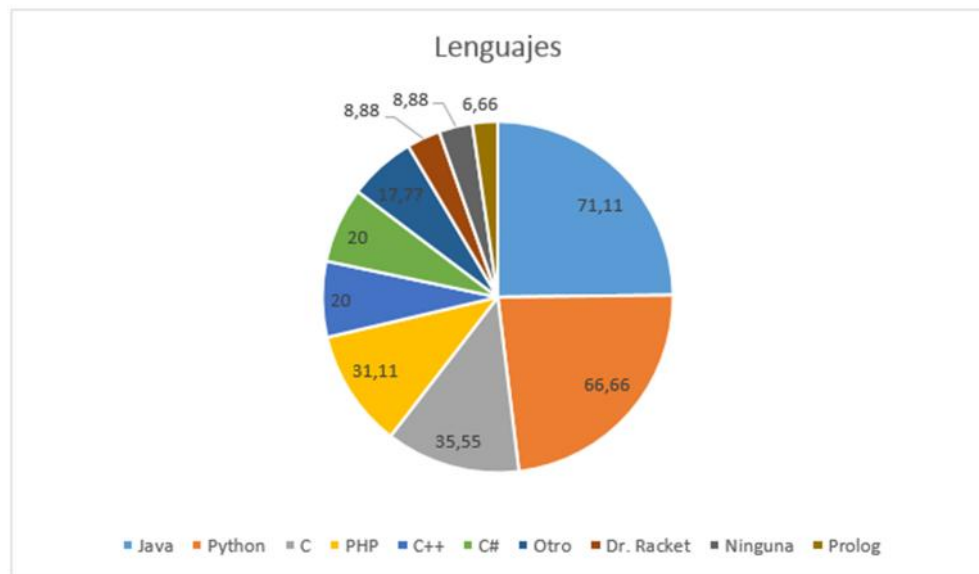
Estudiantes de Cada Jornada



c. ¿Cuáles han sido sus lenguaje de programación preferidos?
(Pregunta de respuesta múltiple, lo que hará que la suma de porcentajes no de 100%)

Dr. Scheme (Dr. Racket)	8.88%
C	35.55%
C++	20.0%
C#	20.0%
Java	71.11%
Python	66.66%
Prolog	6.66%
PHP	31.11%
Otro	17.77%
Ninguno	8.88%

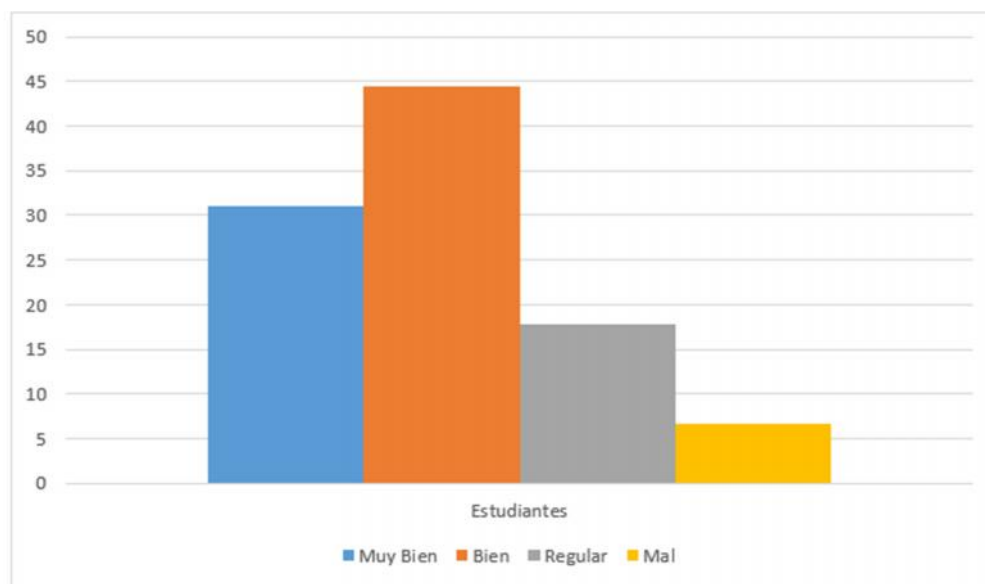
El lenguaje favorito es java con el 71.11% seguido de Python con el 66.66%, C con el 35.55%, PHP con el 31.11%, C++ con el 20%, C# con el 20%, Otro con el 17.77%, Dr. Racket con el 8.88%, Ninguna con el 8.88%, Prolog con el 6.66%



d. ¿Cómo le fue en las asignaturas de programación en la Universidad?

Muy Bien	31.11%
Bien	44.44%
Regular	17.77%
Mal	6.68%
Muy Mal	0%

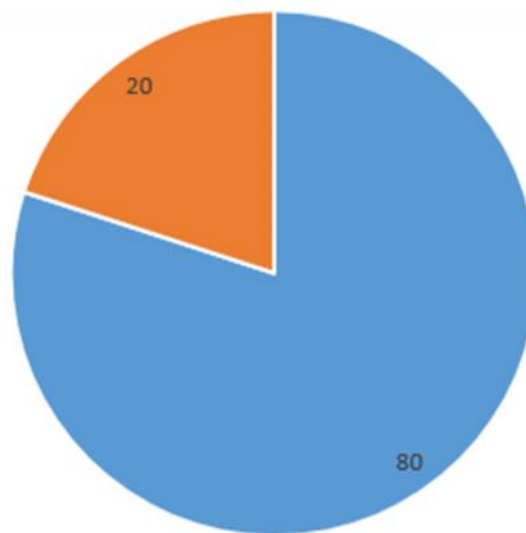
En las asignaturas de programación al 31.11% le fue muy bien, al 44.44% le fue bien, al 17.77% le fue regular y al 6.68% le fue mal.



e. ¿A usted le gusta programar?

Si	80%
No	20%

Al 80% le gusta programar, al 20% no le gusta.

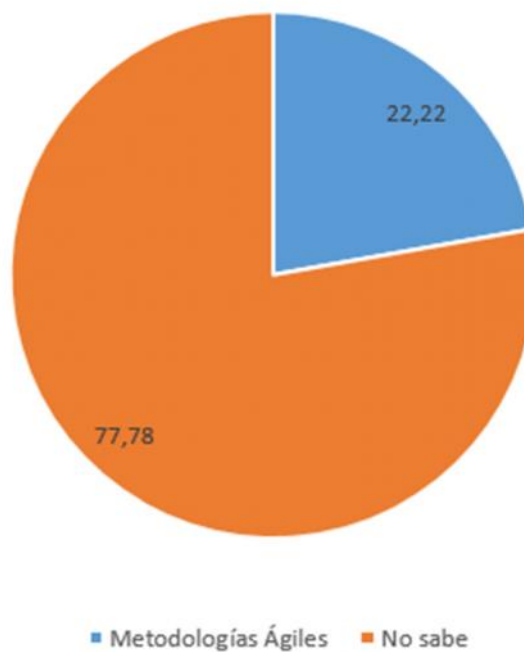


■ Le Gusta ■ No le Gusta

f. ¿Cómo cree que se hace un desarrollo de Software en equipo?
(Pregunta abierta, agrupada después de respondida)

Con metodologías Ágiles	22.22%
No sabe bien como se hace	77.78%

A la pregunta cómo cree que se hace un desarrollo de Software en equipo solo el 22.22% dijo que con métodos ágiles el 77.78% no sabe una manera adecuada de hacerlo.

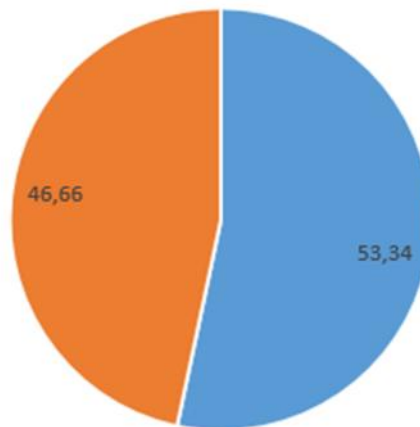


g. ¿Ha escuchado alguna vez de metodologías ágiles para el desarrollo de Software?

Si	53.34%
No	46.66%

A pesar de la contundente proporción de la pregunta anterior el 53.34% dice haber escuchado alguna vez de metodologías ágiles y el 46.66% no la ha escuchado nunca.

¿Han escuchado alguna vez al respecto?

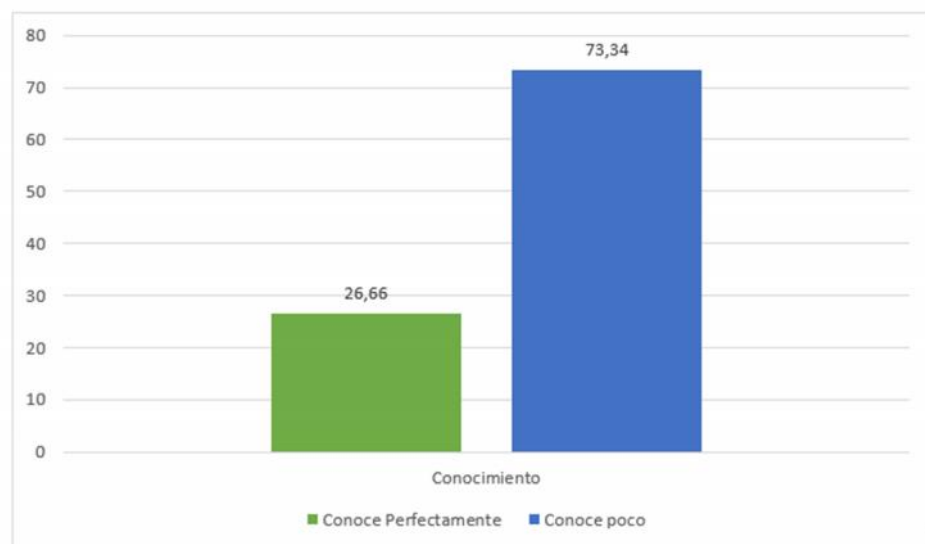


■ Alguna Vez ■ Nunca

h. ¿Que conoce de la metodología Scrum?

Nada	33.33%
Muy poco	8.90%
Medianamente	31.11%
La conoce perfectamente	26.66%

Solo el 26.66% conoce perfectamente la metodología Scrum, los demás conocen muy poco de ella o solo han escuchado algo.



i. ¿En qué área le gustaría ejercer su profesión?
(Pregunta de respuesta múltiple, lo que hará que la suma de porcentajes no de 100%)

Desarrollo de software	75.55%
Analista de sistemas	37.77%
Gerencia de Proyectos de Ingeniería	42.22%
Gerencia de proyectos de software	42.22%
Gerencia de Proyectos de información	31.11%
Crear empresa en desarrollo tecnológico	40%
Consultoría y asesoría en Informática	33.33%
Administración en servicios Informáticos y comunicaciones	33.33%
Desarrollar proyectos de redes de comunicaciones	26.66%
Investigación en ciencias de la computación	33.33%

Al 75.55% le gustaría ejercer su profesión en Desarrollo de software, 42.22% en Gerencia de Proyectos de Ingeniería, 42.22% en Gerencia de proyectos de software, 40% en Crear empresa en desarrollo tecnológico, 37.77% como Analista de sistemas, 33.33% en Consultoría y asesoría en Informática, Administración en servicios Informáticos y comunicaciones, Investigación en ciencias de la computación, 31.11% en Gerencia de Proyectos de información, 26% en Desarrollar proyectos de redes de comunicaciones.



ANEXO B

Encuesta de estudiantes egresados de Ingeniería de Sistemas

Estudiantes graduados del 2010 al 2014 del programa de ISC-UTP, estos son los resultados de las tablas estadísticas logradas con este proyecto de grado:

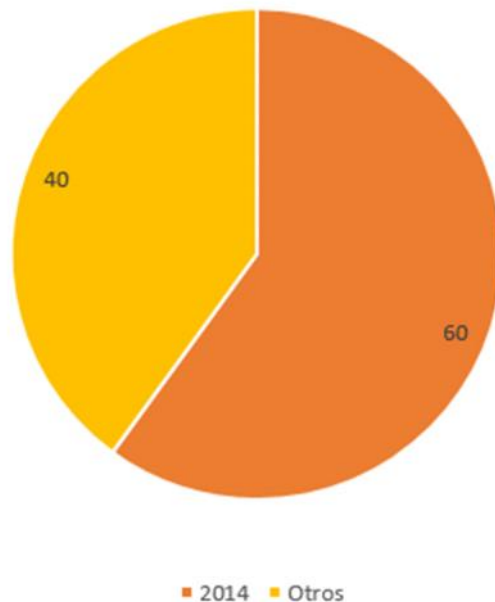
**ENCUESTA A ESTUDIANTES EGRESADOS DE INGENIERÍA DE
SISTEMAS Y COMPUTACIÓN**

Respondida por 10 estudiantes

a. ¿En qué año se graduó?

2010	10%
2011	10%
2012	10%
2013	10%
2014	60%

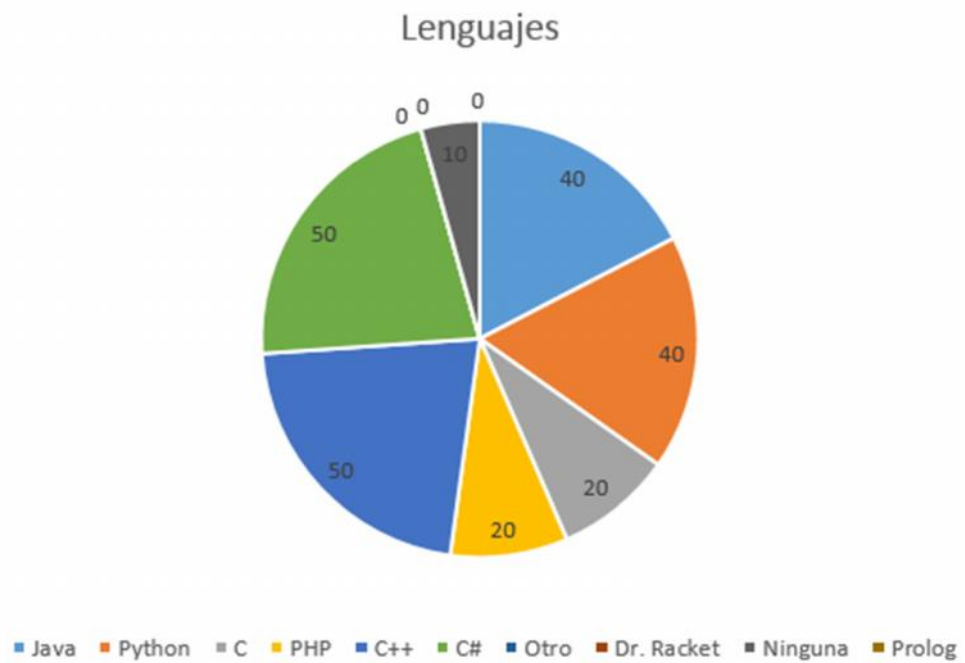
El 60% se graduó en el 2014 y el 40% se graduó entre en 2010 y 2013.



b. ¿Cuáles han sido sus lenguajes de programaciones preferidas?
(Pregunta de respuesta múltiple, lo que hará que la suma de porcentajes no de 100%)

Dr. Scheme (Dr. Racket)	0%
C	20%
C++	50%
C#	50%
Java	40%
Python	40%
Prolog	0%
PHP	20%
Otro	0%
Ninguno	10%

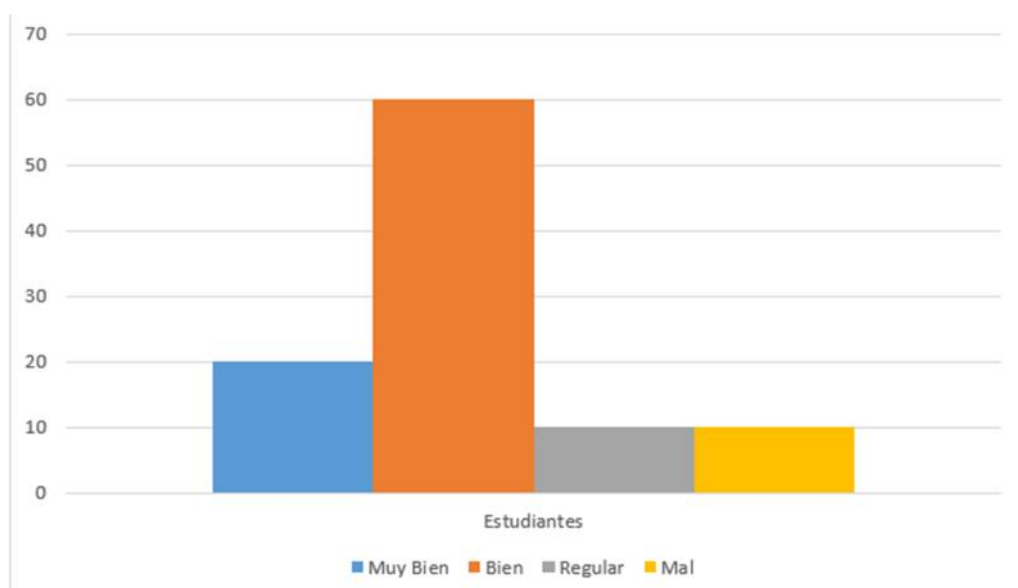
El lenguaje favorito es C++ y C# con el 50% seguido de java y Python con el 40%, C y PHP con el 20%, Ninguna con el 10%, Dr. Racket, Prolog y Otro con el 0%



c. ¿Cómo le fue en las asignaturas de programación en la Universidad?

Muy Bien	20%
Bien	60%
Regular	10%
Mal	10%
Muy Mal	0%

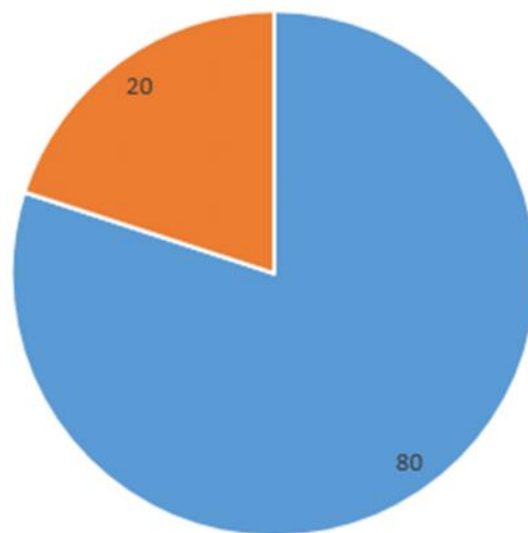
En las asignaturas de programación al 20% le fue muy bien, al 60% le fue bien, al 10% le fue regular y al 10% le fue mal.



d. ¿A usted le gusta programar?

Si	90%
No	10%

Al 80% le gusta programar, al 20% no le gusta.



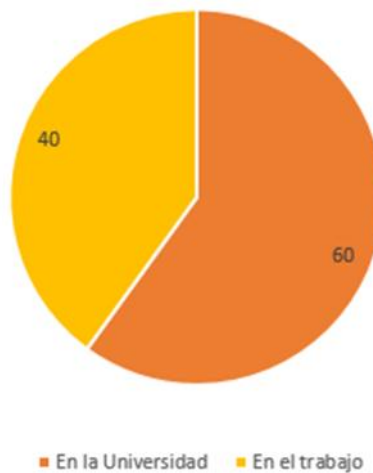
■ Le Gusta ■ No le Gusta

e. Si usted trabaja o ha trabajado en Desarrollo de Software, siente que la Universidad le enseñó la metodología de trabajo en equipo que utilizó en su empresa?

Si	30%
No	70%

A la pregunta: Si usted trabaja o a trabajado en Desarrollo de Software, siente que la Universidad le enseñó la metodología de trabajo en equipo que utilizó en su empresa? solo el 40% siente que lo aprendió en la Universidad y el 60% tuvo que aprenderlo al enfrentarse al trabajo, lo que hace que se confirme el choque que se planteó en el problema inicial del proyecto que es que no tiene la competencia en metodologías ágiles para gestión en desarrollo de software.

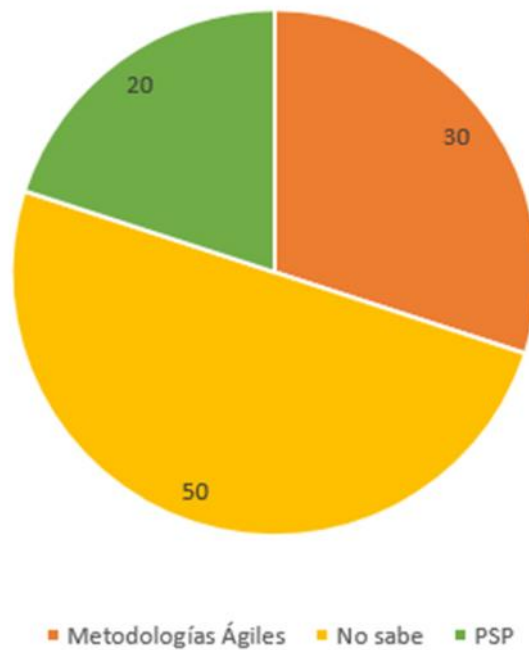
Lo aprendió en:



f. ¿Cómo cree que se hace un desarrollo de Software en equipo?
(Pregunta abierta, agrupada después de respondida)

Metodologías Ágiles	30%
No sabe bien como se hace	50%
PSP	20%

A la pregunta cómo cree que se hace un desarrollo de Software en equipo solo el 30% dijo que con métodos ágiles el 50% no sabe una manera adecuada de hacerlo y el 20% prefiere con PSP que es una certificación en prácticas de calidad en el desarrollo de software que tampoco se enseña en el programa de Ingeniería de Sistemas.

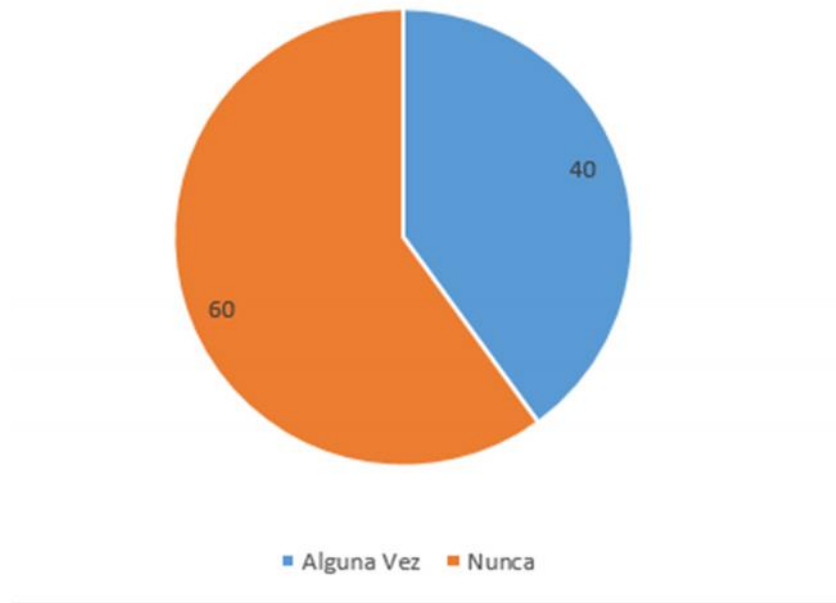


g. ¿Ha escuchado alguna vez de metodologías ágiles para el desarrollo de Software?

Si	40%
No	60%

El 40% dice haber escuchado alguna vez de metodologías ágiles y el 60% no la ha escuchado nunca.

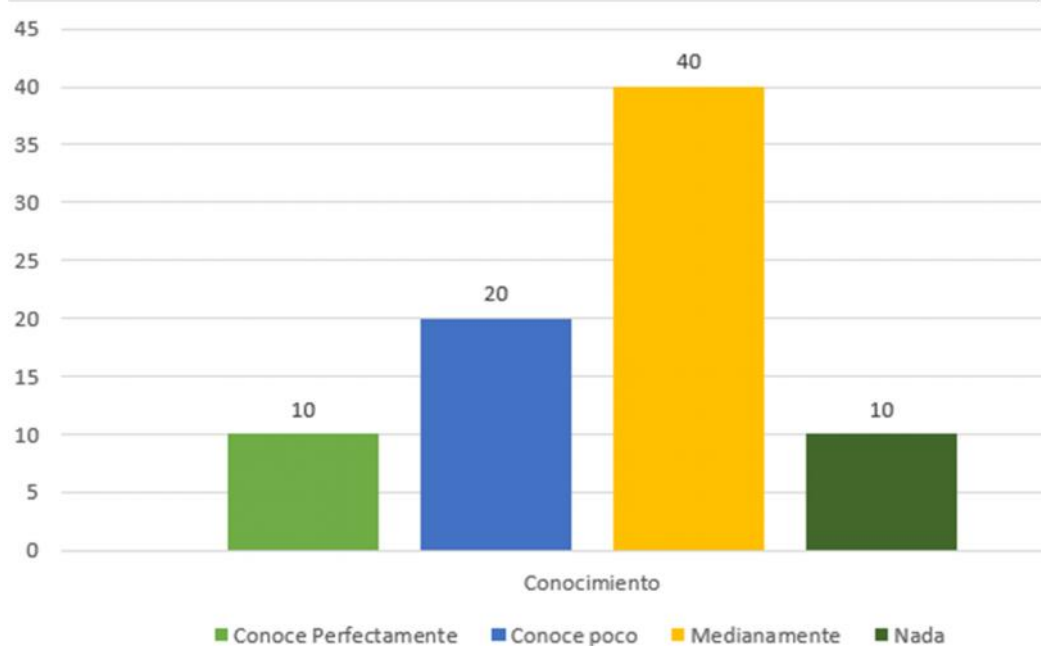
¿Han escuchado alguna vez al respecto?



h. ¿Que conoce de la metodología Scrum?

Nada	10%
Muy poco	20%
Medianamente	40%
La conoce perfectamente	10%

El 40% conoce Scrum medianamente, el 20% muy poco, el 10% no conoce nada y solo el 10% lo conoce perfectamente.



i. ¿En qué área le gustaría ejercer su profesión?
(Pregunta de respuesta múltiple, lo que hará que la suma de porcentajes no de 100%)

Desarrollo de software	50%
Analista de sistemas	30%
Gerencia de Proyectos de Ingeniería	30%
Gerencia de proyectos de software	40%
Gerencia de Proyectos de información	10%
Crear empresa en desarrollo tecnológico	40%
Consultoría y asesoría en Informática	0%
Administración en servicios Informáticos y comunicaciones	20%
Desarrollar proyectos de redes de comunicaciones	10%
Investigación en ciencias de la computación	20%

Al 50% le gustaría ejercer su profesión en Desarrollo de software, 40% en Gerencia de Proyectos de software y Crear empresa en desarrollo tecnológico, 30% en Gerencia de proyectos de Ingeniería y Analista de sistemas, 20% en Administración en servicios Informáticos y comunicaciones e Investigación en ciencias de la computación, 10% en Gerencia de Proyectos de información y en Desarrollar proyectos de redes de comunicaciones, 0% en Consultoría y asesoría en Informática

